Calibration of Expensive Computer Models Using Engineering Reliability Methods

Thesis submitted in accordance with the requirements of
the University of Liverpool for the degree of Doctor in Philosophy

by

Zitong Gong

August 2018

Declaration of Authorship

I, declare that this thesis titled, nd the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

_____

Date:

_____

UNIVERSITY OF LIVERPOOL

Abstract

Institute for Risk and Uncertainty
Faculty of Science & Engineering
School of Engineering

Doctor of Philosophy

Calibration Of Expensive Computer Models
Using Engineering Reliability Methods

by Zitong Gong

The prediction ability of complex computer models (also known as simulators) relies on how well they are calibrated to experimental data. History Matching (HM) is a form of model calibration for computationally expensive models. HM sequentially cuts down the input space to find the fitting input domain that provides a reasonable match between model output and experimental data. A considerable number of simulator runs are required for typical model calibration. Hence, HM involves Bayesian emulation to reduce the cost of running the original model. Despite this, the generation of samples from the reduced domain at every iteration has remained an open and complex problem: current research has shown that the fitting input domain can be disconnected, with nontrivial topology, or be orders of magnitude smaller than the original input space. Analogous to a failure set in the context of engineering reliability analysis, this work proposes to use Subset Simulation—a widely used technique in engineering reliability computations and rare event simulation—to generate samples on the reduced input domain. Unlike Direct Monte Carlo, Subset Simulation progressively decomposes a rare event, which has a very small probability of occurrence, into sequential less rare nested events. The original Subset Simulation uses a Modified Metropolis algorithm to generate the conditional samples that belong to intermediate less rare events. This work also considers different Markov Chain Monte Carlo algorithms and compares their performance in the context of expensive model calibration. Numerical examples are provided to show the potential of the embedded Subset Simulation sampling schemes for HM. The 'climb-cruise engine matching' illustrates that the proposed HM using Subset Simulation can be applied to realistic engineering problems. Considering further improvements of the proposed method, a classification method is used to ensure that

the emulation on each disconnected region gets updated. Uncertainty quantification of expert-estimated correlation matrices helps to identify a mathematically valid (positive semi-definite) correlation matrix between resulting inputs and observations. Further research is required to explicitly address the model discrepancy as well as to take the correlation between model outputs into account.

Acknowledgements

Firstly, I would like to thank my supervisors, Alejandro Diaz and Michael Beer, for their enthusiasm and guidance throughout my PhD studies. I am grateful to China Scholarship Council for the funding of my studentship. I am also grateful for having the opportunity of studying at the Institute for Risk and Uncertainty, University of Liverpool and the Institute for Risk and Reliability, Leibniz Universit´at Hannover. I would also like to thank the following people for their assistance: Arturo Molina-Cristobal and Xin Chen for providing the 'Climb-cruise engine matching' industrial example and data, Vladik Kreinovich and Ingo Neumann for their great assistance when I was visiting Germany studying the correlation. Finally, I would like to thank my family and all my friends both in Liverpool and China for that they support my every decision and let me have a fantastic life.

Contents

List of Figures

List of Tables

Abbreviations

c. o. v.	Coefficient of Variation
CU	Code Uncertainty
DBSCAN	Density Based Spatial Clustering of Applications with Noise
EV	Ensemble Variability
GP	Gaussian Process
HM	History Matching
IDEMC	Implausibility Driven Evolutionary Monte Carlo
k-NN	kth Nearest Neighbor
LHS	Latin Hypercube Sampling
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
MD	Mahalanobis Distance
MD	Model Discrepancy
MMA	Modified Metropolis Algorithm
NROY	Not Ruled Out Yet
OU	Observation Uncertainty
PDF	Probability Density Function
QQ plot	Quantile-Quantile plot
SSHM	History Matching using Subset Simulation
UQ&M	Uncertainty Quantification and Management

Symbols

a	coefficient of Subset-infinity/correlation coefficient
α	candidate movement in MCMC/degree of uncertainty
$\underline{\alpha}$	lower limit for an interval of degree of uncertainty
$\overline{\alpha}$	upper limit for an interval of degree of uncertainty
α_m	midpoint for an interval of degree of uncertainty
$\widetilde{a}_{ij}$	expert-estimated correlations
$a_{ij}^{-}(\alpha)$	lower bound for an interval of fuzzy estimates
$a_{ij}^{+}(\alpha)$	upper bound for an interval of fuzzy estimates
$a_{ij}^{(0)}$	elements of an invalid correlation matrix
A	aspect ratio
$\mathbf{A}$	correlation matrix
AR_W	wing aspect ratio
$\boldsymbol{\beta}$	regression coefficient
BPR	bypass ratio
$c(\mathbf{x}, \mathbf{x}')$	correlation function
d	distance
δ	precision requirement
d_s	standardized residual
D	dimension of variables
Δ	deviation
d_p	unique distance value
$\mathbf{e}_i$	eigenvector corresponding to the smallest eigenvalue
e_{ij}	empirical correlations
e_j	elements in an eigenvector

$\eta(x)$	joint PDF of input variables
EW	empty weight
$E^*(g(\mathbf{x}))$	expectation of the emulation output
$f(x)$	numerical model
F	set of failure scenarios
FFF	Fuel flow factor
FPR	Fan pressure ratio
γ_i	sample correlation factor for level $i-1$
$g(x)$	model output/performance function
$\Gamma(\mathbf{x})$	training data
$h(\mathbf{x})$	regression model
$\hat{g}(\mathbf{x})$	mean value for stochastic model output
$I(x)$	implausibility measure
$\mathcal{I}_F$	indicator function that counts the number of failure scenarios
$\mathcal{I}_I(\cdot)$	indicator function that counts the number of non-implausible samples
J	multiplicity array
k	covariance/number of nearest neighbor samples to a certain sample in DBSCAN
κ	coefficient of Subset-infinity
λ	scaling parameter in adaptive MCMC with optimal scaling/taper ratio/the smallest eigenvalue
λ_i	eigenvalues for a $i \times i$ matrix
L	level
Λ	quarter-chord sweep (degrees)
$\mathcal{L}$	likelihood
m	number of samples when estimating the failure probability
$minPts$	parameter for DBSCAN
n	number of training samples
N	arbitrary number of i.i.d. standard Gaussian variables
$N(\mu, \sigma)$	Normal distribution
N_s	number of seeds for each group in adaptive MCMC with optimal scaling
N_z	Ultimate load factor
n_c	number of chains

n_L	number of samples in each level of Subset Simulation
n_s	number of seeds
OPR	overall pressure ratio
OR	overall drag
$\Phi_q(\mathbf{x})$	space-filling criterion
p	distance norm/dimension of a regression model/path in DBSCAN
P	probability
p_F	failure probability
ϕ	probability distribution
Φ	CDF of the standard Gaussian distribution
p_L	level probability
$p(\sigma^2, \beta)$	Jeffreys prior for stationary variance and regression coefficient
$p(x)$	PDF /
$p^*(x)$	proposal PDF
q	exponent coefficient/dynamic pressure at cruise /
$\rho_L(k)$	correlation coefficient of samples k-steps apart along a chain
R	the number of groups in adaptive MCMC with optimal scaling
s	standard deviation
σ^2	stationary variance
S	set of pairs using exact statistical correlation coefficients
Σ	covariance matrix
$SLST$	sea level static thrust
S_w	wing area
t_c	airfoil thickness to chord ratio
θ	smoothness parameter/input variable in Subset Simulation example
U	Uniform distribution
ε	parameter for DBSCAN/the smallest change for an expert-estimated correlation
$V(x)$	total variance considered for variable x_i
V_c	code uncertainty
V_m	model discrepancy
V_o	observation uncertainty
V_s	ensemble variability
W_{dg}	flight design gross weight

W_{fw}	weight of fuel in the wing
W_p	paint weight
$\mathbf{x}$	samples
ξ	movement of the sample in MCMC
X	set of pairs using expert estimates
y	physical process/system's response value
y^*	specified critical value of a system's response
Y_{SS}	intermediate threshold
z	observation
$Z(\mathbf{x})$	zero-mean Gaussian process

Chapter 1

Introduction

1.1 Background and motivation

Numerical models, also known as simulators, are universally designed and employed to represent and study complex real-world systems. Simulators are used to express physical phenomena in order to understand and predict the behavior of the systems. Sacks et. al. [2] have provided several examples: Nassif. et. al. [3] simulated the delay time of integrated circuits for verification and optimization of the circuit design, Kee et. al. [4] developed a fluid-dynamics model to describe rate constants for estimating the flame velocity, and Wang et. al. [5] modeled the high-temperature combustion reaction for alkane compounds.

In addition, physical analogs of some computer experiments are infeasible, not only in terms of time but also in terms of practical condition or even security problems. For example, to predict the ambient temperature for the next 100 years or to estimate the radioactive release in a nuclear disaster. In such cases, simulators are more needed.

However, no model is perfect. There always exists an error that is the difference between the model prediction and the actual physical quantity when building the model with particular assumptions, simplifications or approximations [6]. For a simple example, in structural engineering, the mean weight for a certain batch of beams of the same size and material follows a Normal distribution (Central Limit Theorem [7]). The parameters

(mean and variance) included in this mathematical model (the Normal distribution) are unknown and need to be estimated by making measurement of limited samples randomly selected from the population. The weight of any individual beam is then considered as the same as the estimated mean weight. The modeling here is a proper representation of the reality while allowing a proper level of uncertainty—it is unnecessary to measure the weight for a large number of beams one by one when they are from a same batch. In a word, a computer model always inherently carries uncertainty margins because of limited resource, mainly in terms of time.

A trustworthy model should be a representation of the corresponding physical system, especially in the field of risk assessment when modeling safety performance metrics. Several authors have discussed the concept of uncertainty for building a model, see e.g. Beck [8], Gallegos and Bonano [9], Zio and Apostolakis [10], Nilsen and Aven [11], and Helton et. al. [12]. In synthesis, the uncertainty arises due to:

1. Experimental error that occurs in the execution of experimental designs, which includes human mistakes, instrument bias, random error caused by environmental conditions or other unpredictable factors.

2. Poor understanding of the physical system, which causes the inadequacy of simulators or even wrong simulators.

3. Parameter uncertainty, which is the inability to describe the inherent variability of the model parameters. This if often caused by the lack of data or the change in environmental conditions of the physical system under study.

4. Additional simplification to translate a physical system, e.g. flatten a 3-D continuum to a 2-D model to focus on planar characteristics.

5. Approximate implementation of numerical methods using computer codes, e.g. optimization problems, evaluating integrals, or differential equations, where the analytical solutions are often not available due to finite computing time.

6. Stochastic systems not always produce the same output for a given input, which is unpredictable. A few components that can be stochastic in nature include random

input variable, random time-delays, and stochastic dynamic processes, which is a dynamic system (described by a function dependent on the time and a geometrical space) subjected to random noise.

Different activities are required to quantify the uncertainty of building a model, including verification, validation and calibration [13]. According to the Guide for Verification and Validation in Computational Solid Mechanics [14], the definitions for verification, validation and calibration are as follows.

Definition 1.1. *(Verification)* The process of determining that a computational model accurately represents the underlying mathematical model and its solution.

Definition 1.2. *(Validation)* The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model.

Definition 1.3. *(Calibration)* The process of adjusting physical modeling parameters in the computational model to improve agreement with experimental data.

Verification [15] can be conducted by comparing computational implementation with the conceptual model to prove the rationality of the implementation. Validation [16] can be conducted by comparing model outputs with observational values to quantify the uncertainty of the model output. Similarly with validation, calibration [17] can be conducted by comparing model outputs with observational values as well, but to make inference about unknown parameters of the model. The process of model development needs an iterative refinement through verification, calibration and validation until the model accurately represents the underlying scientific phenomenon [18]. A workflow of the model development process is shown in Figure 1.1.

Among above activities of uncertainty quantification, this thesis focuses on a specific form of model calibration, which assesses model input space governed by observations of physical quantities. The calibration process of using observed data to make adjustments to model inputs is vitally important wherever a computer model is used. The prediction ability of computer models relies on how well they are calibrated to experimental data.

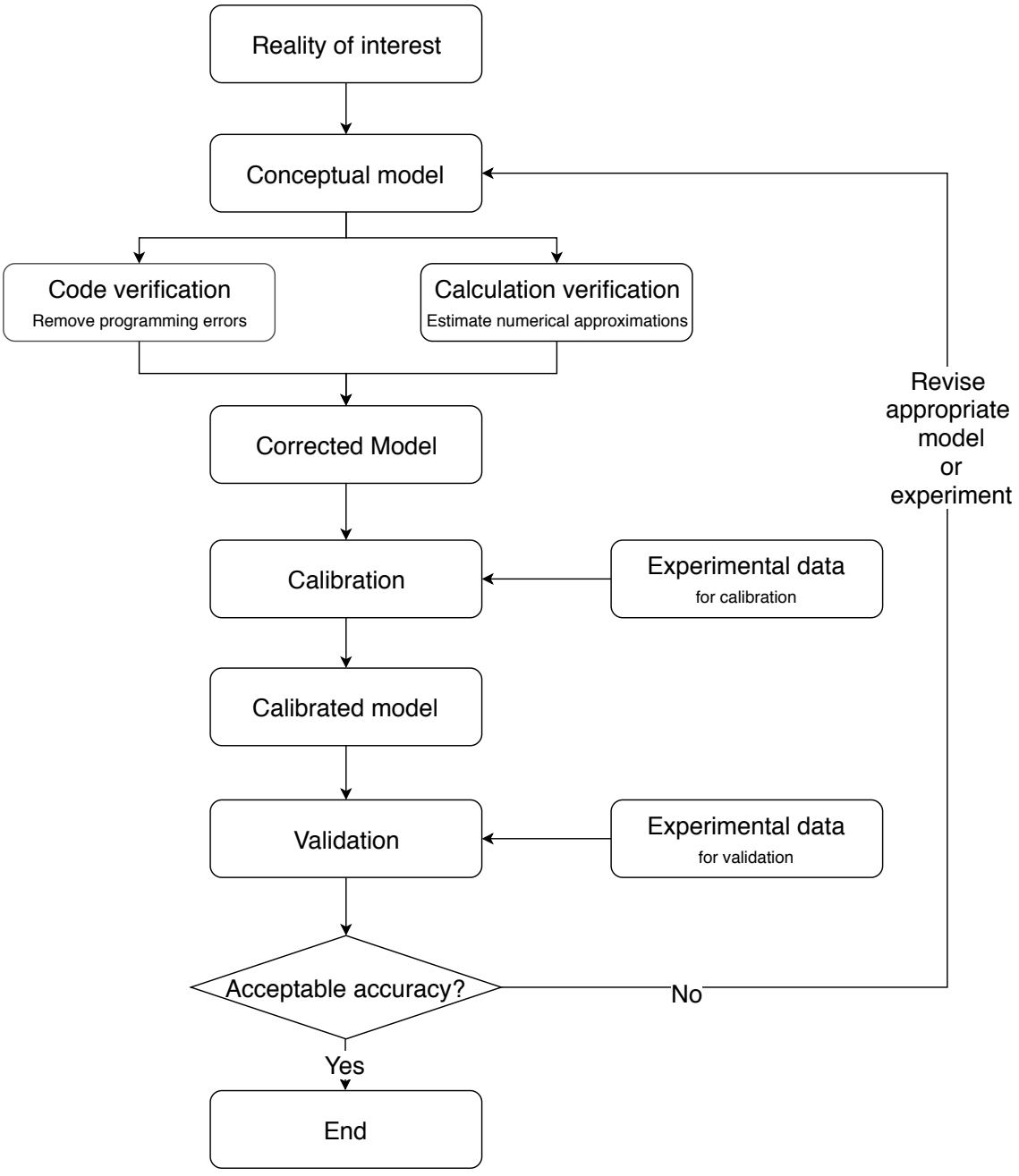


FIGURE 1.1: The workflow of model development based on the Guide for Verification and Validation in Computational Solid Mechanics.

Calibration is the inverse process of building a simulator. However, for complex simulators, computing input values given observations is analytically intractable. In practice, there can be too many solutions for an observation. For example, an aircraft's wing weight is determined by many parameters, such as the wing area, the weight of fuel, the aspect ratio, etc. If these parameters are fixed, the wing weight takes a known value. However, with a given value of the wing weight, there can be many parameter configurations to reproduce this value. A large wing area with a low weight of fuel can produce the same wing weight value as a small wing area with a large weight of fuel. Similarly with all other parameters. Hence, calibrating a simulator is done through sampling. The steps are as follows [19]:

1. Identify a calibrating range of model inputs;
2. Generate samples within the calibrating range;
3. Assess that how well model outputs match given observations;
4. Adjust inputs.

In this way, the input domain that provides a match between model outputs and observations is eventually traced down from their correspondingly qualified outputs. Evaluating whether model outputs match the observations is done through goodness-of-fit.

Classical goodness-of-fit methods include acceptable windows, minimizing deviations and likelihood functions. Acceptable windows (also called tolerance interval) [20] give an acceptable upper bound and a lower bound for each observation. As long as the model output is within the acceptable window, it is said to be a match for the observation. Minimizing deviations calculate the distance between model outputs and observations, then select the outputs with minimum deviations. This approach captures the magnitude of deviations. An example is the method of least squares [21]. Likelihood functions [22] select inputs that can maximize the probability of reproducing the observations. It requires probability distribution functions of model inputs.

However, simultaneously to the improvement of the computational power during the last decades, the complexity of computer models has also increased. Complex engineering

systems are usually investigated by computationally expensive computer models. This means that to conduct any type of analysis that takes many model runs (e.g. model calibration, sensitivity analysis or reliability analysis), a considerable amount of resources such as processing time and memory are needed. For example, a single run of a finite element model can take hours on a desktop [23–25]. If it is ran on a high performance computer, the computation would be faster. Nonetheless, a finite element model can be refined to a point even high performance computers take a long time to run. For the purpose of Monte Carlo simulations, the number of samples grows exponentially with the number of dimensions. In such cases, computationally expensive models often hinder the calibration process and affect the efficiency of above goodness-of-fit methods.

Early works [2, 26–28] were concerned with selecting a smaller sized sample of input configurations to run the expensive model, and based on which estimating/interpolating the output at other input configurations. Kennedy and O'Hagan [29] employed a Gaussian process emulator [30] to model a complex simulator's output. The Gaussian process emulation used here is a method of interpolation. It adopts the principles of Bayesian predictive inference established by Aitchison and Dunsmore [31]. It assigns prior distributions for the calibrating parameters then derives the posterior distributions given benchmark or training samples. A relatively small amount of benchmark samples are evaluated by the simulator. The rest of samples are evaluated by the emulator that is much cheaper with respect of run time. In addition, importantly, Gaussian process emulator produces confidence intervals for the output predictions. Generally, if samples evaluated by the emulator are more close to the benchmark samples evaluated by the simulator, their outputs are more certain due to the interpolation nature of this approach.

Following the above work [29], Kennedy and O'Hagan [32] presented a Bayesian calibration technique attempting to correct the model discrepancy that is the inadequacy of a model. Another Gaussian process emulator, independent from the one used for modeling the simulator, is employed to model the inadequacy of the simulator. This has been widely applied in many fields: thermal challenge problems [33, 34], water quality management [35, 36], aerodynamic flow turbulence simulations [37] among other

applications.

However, Bayesian model calibration yields a posterior distribution no matter whether there indeed exists a match between model outputs and observations. When observing data is very expensive, only one or a few observations are available, which might not provide enough information to construct a posterior distribution of the model parameters given the data. History matching (HM) [38] is a calibration technique that can solve this problem. HM is a form of model calibration that iteratively cuts down the initial input space to focus only on the subset of the input space that is likely to provide a plausible match between output values and observed data.

The aim of HM is to calibrate complex and costly simulators. In order to reduce the computational cost, Gaussian process emulators are an important and necessary step in HM. The workflow will be explained in section 2.3. The accuracy of the current iteration depends on the accuracy of the current emulator, which can be improved in sequential iterations by adding more training data. Note that adding more training data and refitting the emulator only focus over the currently potential target space. The potential target space is termed 'non-implausible' domain in HM, which is assessed by an implausibility measure of HM that can comprise all sources of uncertainty. Non-implausible domain means that it can provide a match between model outputs and observations according to the current accuracy. When the accuracy of the emulator is improved in sequential iterations, the current non-implausible domain has the possibility to be determined a implausible (which can no longer provide a match). At each iteration, HM rules out samples in the implausible input domain and generates samples that eventually define the fitting input domain providing a reasonable match between the model output and experimental data.

HM was first applied by Anterion [39] to the oil industry to diminish computational time in inverse problems for reservoir simulations. It was used to adjust a model of an oil reservoir until the model outputs reproduced/matched the historical observations: production and pressures. The geological properties as model inputs were adjusted to improve the match. The term 'history' does not constrain HM to time series analysis. Once the model has been history matched, it can be used to simulate future reservoir

with a higher degree of accuracy. Since then, more applications using HM to obtain reservoir characteristics have emerged [40–42]. Recently, HM has also been successfully applied to highly nonlinear geophysical simulators [43], to an energy balance climate model [38], to galaxy formation models [44, 45], and to large climate systems modeling [46] amongst other applications. The iterative structure of HM enables it for calibrating complex, high dimensional simulation models.

The advantage of HM over different forms of calibration is that it can diagnose that a model is not adequate or as to say the observation has no match with the model output by producing an empty input domain. This is done through determining the whole input domain as implausible by the implausibility threshold, whereas traditional maximum likelihood methodologies or the Bayesian model calibration always yield an input domain by the likelihood nature. Figure 1.2 gives a simple illustration. The observation is denoted as the green line with the measurement uncertainty (experimental error), represented by dashed green lines. One possible model for a physical process is visualized as the gray curve, and let's suppose another model that is inadequate corresponding to the blue curve. If the model with the blue curve is under calibration, HM would result that there is no fitting input according to the implausibility measure, while likelihood based methods would result in an interval with a large probability to match the observation according to the maximum likelihood principle, however the correct fitting input with observation uncertainty should be the red interval. This example illustrates that by producing an empty resulting input domain HM is capable of pointing out an inadequate model. But conversely, if HM does detect a fitting input domain, it not necessarily means that the model is adequate. This example also illustrates that HM has conceptual difference from likelihood based calibration methods. It discards implausible input areas, rather than making probabilistic statements about the most likely input values given observed data [38]. This conceptual difference is also the reason of HM can find the fitting input domain given even only one observation.

Despite the advantages of HM, one of the most challenging problems in implementing it is the efficient generation of samples in the set of interest, on which the model is rerun at every iteration. For a certain iteration, the set of interest can be disconnected,

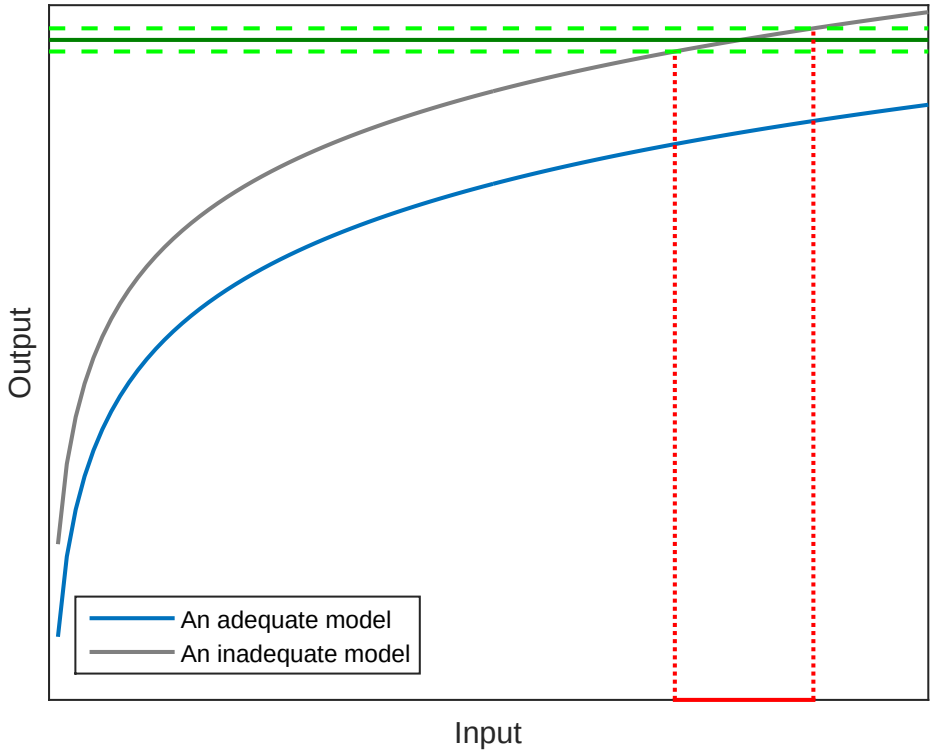


FIGURE 1.2: A comparison of resulting input from HM and likelihood based methods when the model is inadequate. HM can diagnose that a model is inadequate by resulting to an empty input space while likelihood based methods always yield to a 'fitting' input domain due to the probabilistic nature even the model is inadequate.

of nontrivial topology or be a very small fraction of the initial input space. To obtain enough samples on the set of interest, millions of samples need to be generated on the initial input space. This will be illustrated later in detail by a 10-D wing weight model (subsection 4.5.2). Here a simple picture (Figure 1.3) is presented to illustrate the reason why non-implausibility sampling can be a low-probability problem. A model output is denoted by a blue curve. If a few observations (denoted by dark green lines) are given with large measurement uncertainty (denoted by dashed green lines), the fitting input domain is relatively large (25% of the input space, shown in red intervals). If only one observation (denoted by a black line) is given with a small measurement uncertainty (denoted by gray lines), the fitting input domain is relatively small (5% of the input space, shown in a light blue interval). A high-dimensional input space would make the non-implausible domain even more difficult to be found. In the problem of calibrating the 10-D wing weight model mentioned above, the non-implausible probability is to

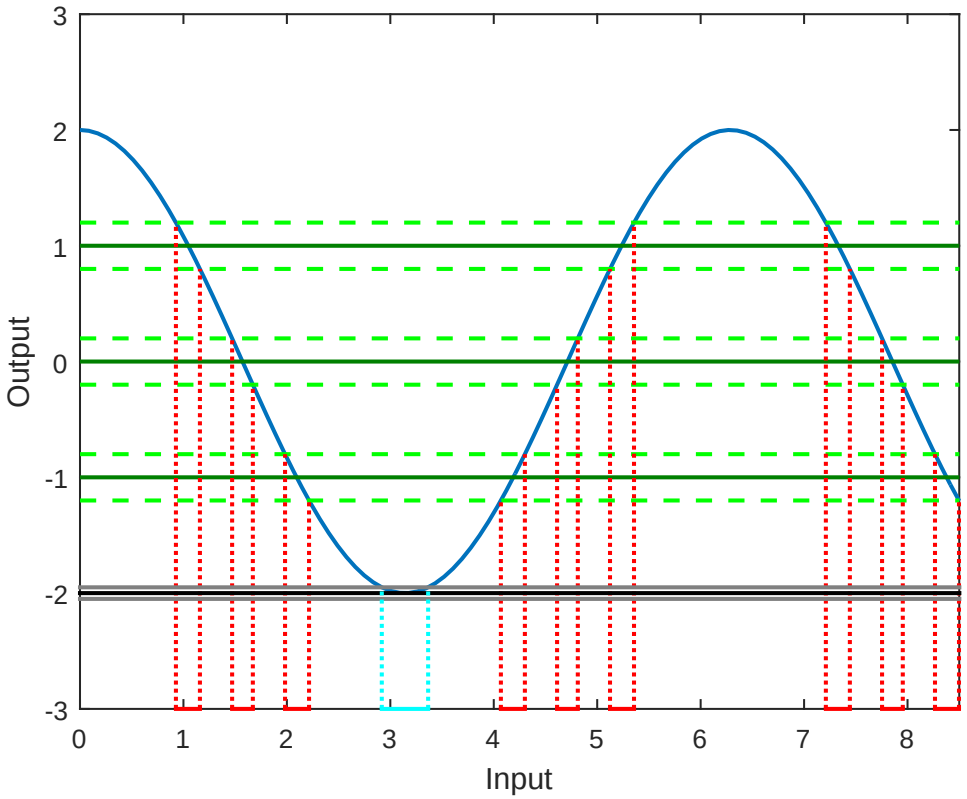


FIGURE 1.3: An illustration for non-implausibility of low-probability. For this 1-D input model, if a few observations (denoted by green horizontal lines) are given with large measurement uncertainty, the non-implausible domain can be relatively large, 25% of the input space, shown in red intervals. If only one observation (denoted by a black horizontal line) is given with small measurement uncertainty, the non-implausible domain can be relatively small (5% of the input space, shown in a light blue interval).

10^{-6}. This makes efficient sampling extremely challenging. Currently, there are at least three classes of methods to generate samples in the non-implausible set: an acceptance-rejection strategy [45], an implausibility driven evolutionary Monte Carlo algorithm [47], and a perturbation approach [48], however, their efficiency depends on several assumptions or requirements (more details will be provided in chapter 2). Clearly, the efficient generation of samples only in the set of interest represents one of the main intellectual challenges to be solved before HM can become a more widely accepted methodology for calibration.

On the other hand, HM assesses model input space governed by observations of physical quantities, namely, it explores the relationship between inputs and observations. Thus, based on HM, correlation among the resulting inputs and the observations can be studied. Often for HM one or only a few observations are needed, hence there are not enough

data to calculate the correlation. A natural idea to supplement this situation is to rely on expert estimates. Expert estimates can be crude. Hence the other challenge is to quantify the uncertainty of the expert-estimated correlation matrix. Sometimes, experts provide us with not a numerical estimate for the correlation coefficient but interval estimates that contain the expert's confidence levels. Experts often express their confidence levels in terms of fuzzy words such as 'about' or 'close to', therefore it is reasonable to used the techniques of fuzzy expressions [49–51]. Fuzzy set theory relaxes the need for precise estimates. Analyst can express the available information via a series of nested intervals. In this case, quantifying the uncertainty of the correlations estimated with fuzzy intervals can be more complicated and worth studying.

1.2 Aim and objectives

The aim of this thesis is to solve a challenging sampling problem for HM, making it a more suitable model calibration approach for multi-dimensional, complex, expensive simulators. The simulators studied in this work are focused on aerospace engineering. However, the methodology proposed is applicable in different fields, where model calibration is needed, such as in [38, 40–46]. The requirement for sampling only in the set of interest can be circumvented by the use of engineering reliability methods. Because HM is expected to find the non-implausible domain for a very small number of observations, this non-implausible domain can reduce to be orders of magnitude smaller than the original input space. Therefore, the non-implausible domain can be interpreted as a failure set in the engineering reliability context, where sampling is done through rare event simulation. For this reason, this work adapts Subset Simulation [52]—a widely used technique in engineering reliability computations—to the sampling procedure for HM. Subset Simulation is also able to sample from model input domain that is disconnected or has a complex topology. This is a desirable property for sampling the reduced non-implausible domain in HM, which will be illustrated in section 4.4. Subset Simulation has variants of sampling schemes. At its core, Subset Simulation relies on a Markov Chain Monte Carlo scheme. Throughout the years, several samplers have been

proposed. This work makes a comparison of their performance differences in the context of HM.

This work also explores a further improvement of the proposed method, History matching using Subset Simulation (SSHM). SSHM involves iteratively updating the emulator so that it could capture the simulator's details in the region of interest. Because HM focuses on the non-implausible domain, the non-implausible samples at the current iteration can be recycled. A classification method is applied to select the updating data directly from those non-implausible samples—DBSCAN ([53]) is used to identify all clustered non-implausible samples that located on disconnected regions, so that each region can be ensured to have new training data to improve the local accuracy of the emulator.

In order to verify that the proposed method is capable of solving real world problems, SSHM is applied to an industrial engineering problem: 'Climb-cruise engine matching'. This was one of the challenges regarding to 'uncertainty quantification and management' proposed by Airbus. The parameter space for aircraft designs needs to be calibrated in the context of a 24-hour operation aircraft ruled by noise curfews and gas emission controls. Hence the fitting design parameters for an aircraft can be determined by given targets of noise values and gas emission values.

The results provided by HM can be further analyzed. HM finds the fitting input domain which can provide a match between input parameters and observations. In this respect, the relations between model inputs and observations are worth exploring. In the case that there is not enough data to statistically compute a correlation matrix for the resulting inputs and given observations, it is natural to rely on expert's estimates. However estimating the correlation coefficient respectively may violate the mathematical meaning of a correlation matrix. Therefore, a method of quantifying the uncertainty of expert's estimates by quantifying the smallest change to bring back a valid correlation matrix is proposed.

To fulfill the aim, the corresponding objectives are as follows:

1. Study the potential of applying Subset Simulation to implausibility based sampling problems and then solve the problem of sampling in the non-implausible domain

through Subset Simulation;

2. Compare different sampling schemes within the newly proposed algorithm;

3. Extend the development of the proposed algorithm, such as the classification of training samples that belong to disconnected non-implausible input domain in order to cope with disconnected non-implausible domains;

4. Apply the proposed algorithm to an engineering problem in an industrial setting;

5. Identify the invalid expert-estimated correlation matrix, which does not meet the mathematical requirement of a correlation matrix—being positive semi-definite;

6. Establish the relation between being positive semi-definite and the corresponding change for an invalid correlation matrix;

7. Quantify the uncertainty of the estimated correlation matrix by determining the smallest possible change of individual correlation coefficients in order to make a positive semi-definite correlation matrix.

The examples used in this work are listed as follows:

1. A numerical example of Subset Simulation will be presented. It is a frame structure in its origin field of structural engineering to calculate the failure probability.

2. Two numerical examples will be presented to visually show that Subset Simulation is capable of sampling from non-implausible domains disconnected or with complex topologies.

3. A 1-D numerical example to illustrate the implementation of the proposed method will be presented.

4. A 10-D wing weight model with meaningful variables in reality to illustrate the implementation of the proposed method will be presented.

5. A 2-D numerical example to visualize the advantage of classified training data will be presented.

6. A real-world industrial engineering problem 'climb-cruise engine matching' will be analyzed using the proposed method.

7. Based on the HM result of the 'climb-cruise engine matching' problem, an example to illustrate the uncertainty quantification of the expert-estimated correlation matrix will be presented.

1.3 Overview of the remaining chapters

This dissertation is organized as follows.

Chapter 2 gives the basic calibration framework of HM. It presents important components of HM and the uncertainty considered in the process of HM. It also points out the challenge in sampling for HM—sampling should only focus on the non-implausible domain, however which can be disconnected, of non-trivial topology and orders of magnitude smaller than the calibrating input space.

Chapter 3 reviews Subset Simulation, a widely used method to sample rare failure events and compute failure probability for engineering reliability analysis. This chapter provides the details of Subset Simulation and discusses different sample-generating algorithms of Subset Simulation.

Chapter 4 provides procedures and examples of adapting Subset Simulation with HM. Benchmark criteria are introduced to evaluate different sample-generating algorithms of Subset Simulation within the framework of SSHM. This chapter also proposes to apply a classification method when adding training data for each iteration, which ensures that disconnected non-implausible domain can have evenly spread new training data, so that the emulator on the corresponding domain can be updated. (Objectives 1,2 and 3 are addressed in this chapter.)

Chapter 5 uses the proposed SSHM to solve an industrial problem. It illustrates that SSHM can quickly explore and reduce input parameter space with given observations for a high dimensional and complex simulator. (Objective 4 is addressed in this chapter.)

Chapter 6 explores the correlation between model inputs and observations. In this chapter, the uncertainty of an expert-estimated correlation matrix is quantified in the case that the correlation matrix can not be statistically calculated due to not enough data. The algorithms of this uncertainty quantification are provided for both numerical estimates and fuzzy estimates of a correlation matrix. (Objectives 5, 6 and 7 are addressed in this chapter.)

Chapter 7 gives conclusions and areas for future research.

Chapter 2

History Matching

2.1 Introduction

The reliability of computer models depends on how well they are calibrated to experimental data. Two major issues often hinder the calibration process: first, a complex simulator may take a large amount of time for a single run [25], which makes the computation of the model output infeasible for a large number of samples covering the input space. Second, a high-dimensional input space may result in a fitting input domain only of a small fraction of the initial input space given one observation, which makes the sampling very challenging.

History matching (HM) [38, 54] is a calibration method, which sequentially cuts down the input space to find the fitting input domain that provides a match between model outputs and observations. HM solves the first of the above problems by using an emulator [30] to approximate the simulator's output based on a relative small number of simulator evaluations, known as training runs or training data. HM solves the second of the above problems by only focusing on the non-implausible domain. It defines an 'implausibility measure', which takes into account a variety sources of uncertainty, to compare the model output with the observation and divide the input space into implausible and non-implausible. At each iteration, the implausible domain is discarded and the non-implausible domain is refocused as the potential fitting input domain that may provide

the desired match. Then new training data are added at each iteration to update the emulator, so that the estimation of the model output is improved with generally smaller uncertainty. In the end, a fitting input domain, orders of magnitude smaller than the initial calibrating space, can be identified by iteratively excluding the input space of low plausibility.

As mentioned above, HM is capable of assessing a variety sources of uncertainties by including them in the implausibility measure. It is also capable of using only one observation or only a few observations to seek the fitting input domain. Additionally, it is capable of diagnosing a model by producing an empty input space, pointing out that the model is incorrect.

'Non-implausible' domain means that it is not ruled out yet according to the current emulator's accuracy but it has the possibility to be ruled out in sequential iterations. This is also as known as 'not ruled out yet' (NROY) space [47]. The term 'Wave' [38] is used in HM to represent each iteration, which comprises sampling the NROY input space, adding simulation data, re-emulating within the NROY space and removing the samples with large implausibility. This iterative process is also termed refocusing [45].

Gradually reduced input space produces a higher density of emulation runs at each wave, so that more non-implausible samples tend to be detected. In addition, the simulator may behave smoother on a reduced space hence easier to emulate. However, this continued refocusing brings the complication of evaluating the implausibility for each sample at each wave. Hence, an efficient sampling method is needed to screen the input space [45], which is the core problem solved by this dissertation.

2.2 Uncertainty Quantification

One of the convenient properties of HM is that it can include a variety sources of uncertainties when matching observations. This is important in case some fitting samples are excluded due to experimental measurement uncertainty, computational approximation, inadequacy of the model, stochastic variability or other unknown factors of uncertainty. Andrianakis et. al. [38] presented a way to to quantify the uncertainty of HM via:

observation uncertainty (OU), code uncertainty (CU), model discrepancy (MD), and ensemble variability (EV). They are explained separately as follows.

2.2.1 Observation uncertainty

Observation uncertainty (OU), denoted by V_o, represents the experimental uncertainty when measuring data. Historical data measured by experiments in the past have OU as well. OU is subject to the finite accuracy of measurement instruments, human operations and environmental conditions, such as precision error, unit conversion error and instrument bias. Observation uncertainty can be divided into two parts. If one experiment is conducted using the same instrument in the same way and in the same conditions, there occurs a systematic error of the same value. Systematic error can be reduced by improving the design of the experiment or using an instrument with higher accuracy. There also exist another part of the observation uncertainty, as known as a random error that may vary from observation to another. For example, to the extremes of the accuracy limits, the last significant digit of a number may be read differently at each time, even by a same person.

Let z_i, $i = 1, ..., r$ denote measurements and let z_{i0} denote the actual values of the physical quantities under study. There always exists a measurement error $\Delta z_i \stackrel{\text{def}}{=} z_i - z_{i0}$. For example, a gauge accurate to 0.1 units means that the 'true' z_{i0} should be within the distance of $|\Delta z_i|$ for the observation value z_i. OU is calculated by $\sqrt{V_o} = |\Delta z_i| = \frac{0.1}{2} = 0.05$, since the observed value could be either sides of true value. In practice OU is quantified by collecting a set of samples and computing the samples' variance.

2.2.2 Code uncertainty

Code uncertainty (CU), denoted by V_c, represents the uncertainty imposed by the fact that an expensive simulator cannot be run everywhere on the input space. HM makes the calibration of complex models feasible because it reduces a larger number of simulator evaluations by using a statistical model to substitute the simulator's output for a large number of samples. The substitution model is called an emulator [30], which generally

runs much quicker than a complex simulator. Because an emulator gives a predictive posterior distribution of the model output given simulation data, CU is automatically quantified through the variance of the emulator's output. The variance of the emulator's output is various on different samples. The variance is generally small when the sample is close to training data due to the interpolation nature of the emulator. The maximum value of the emulator's variance $max(V_c(\mathbf{x}))$ is used as one of the stopping criteria for HM (more stopping criteria are listed in section 2.8). If $max(V_c(\mathbf{x}))$ is smaller than all other uncertainties $V_o + V_m + V_s$, it means that CU is not the main source of uncertainty, thus updating the emulator in further waves barely changes the implausibility cut-off.

2.2.3 Model discrepancy

Model discrepancy (MD), denoted by V_m, stems from the inability of perfectly modeling a physical system. It is also called model inadequacy, model error, model form error, model bias or structural uncertainty. In practice, complex physical systems are always simplified by computer models. Besides, there often exist unknown factors that are supposed to be considered into the model, but they are ignored due to the incomplete understanding of all aspects of the physical system. In fact, no model is perfect, as a aphorism saying by George Box, 'all models are wrong, but some are useful' [55]. All models have discrepancies between output predictions and observed data.

Many authors [32, 56, 57] model the MD via a zero-mean Gaussian process (the definition and more details of a zero-mean Gaussian process (GP) will be discussed in section 2.5), which specifically takes the form: $\delta(x) \sim GP(0, \sigma^2 c(x, x')|\theta)$, where the variance $\sigma^2 c(x, x')|\theta$ takes in charge of correcting the model discrepancy. A Bayesian approach is used to estimate the unknown parameters. Prior distributions are assigned to them then updated by observations to have a posterior distribution, of which the process involves Markov Chain Monte Carlo (MCMC) techniques. The modeling of MD is an active research area, important questions remain about the choice of design points, the limitation of physical observations, the approximation of high dimensional quadrature [32] and other extensive aspects. In the context of HM, Andrianakis et. al.

[48] simplifies the model discrepancy to a constant value over the whole input domain specified by the model analyst, which is adopted in this work.

2.2.4 Ensemble variability

Ensemble variability (EV), denoted by V_s, comes from stochastic simulators. As opposed to deterministic simulators, outputs of stochastic simulators may differ at each run. A stochastic simulator traces the evolution of components that can change randomly. Such components with stochastic nature are e.g. random input variable, random time-delays, and stochastic dynamic processes, which is a dynamic system (described by a function dependent on the time and a geometrical space) subjected to random noise. The output of a stochastic simulator is computed affordably multiple times to estimate a mean value $E(\hat{g}_j(\mathbf{x}))$ and EV is the variance of the output:

$$V_s = E(\hat{g}_j(\mathbf{x}) - E(\hat{g}_j(\mathbf{x})))^2 \tag{2.1}$$

For example, a biological model is stochastic when (the population of a molecular species is low and) the number of the molecular is considered as a Gaussian random variable [58]. In this work, it is assumed that the simulators are deterministic, for which V_s is taken as zero.

2.2.5 Uncertainties in HM

The uncertainties in HM are the sum of all above uncertainties $V(\mathbf{x}) = V_o + V_s + V_c(\mathbf{x}) + V_m$. They are fundamental components of HM, as shown by Figure 2.1. Underestimating OU may lead to biased resulting inputs since a deviated observation is targeted. Along the iterations of HM, an emulator is gradually updated with smaller CU expected in the implausibility measure, which makes the calibration more precise. Simulators with large MD, EV or overfitting give rise to a mismatch between historical data and future data.

To gain more insight, a special case to assemble the above sources of uncertainty is presented. An observation z can be represented by a fitting input $\mathbf{x}^*$ evaluated by a

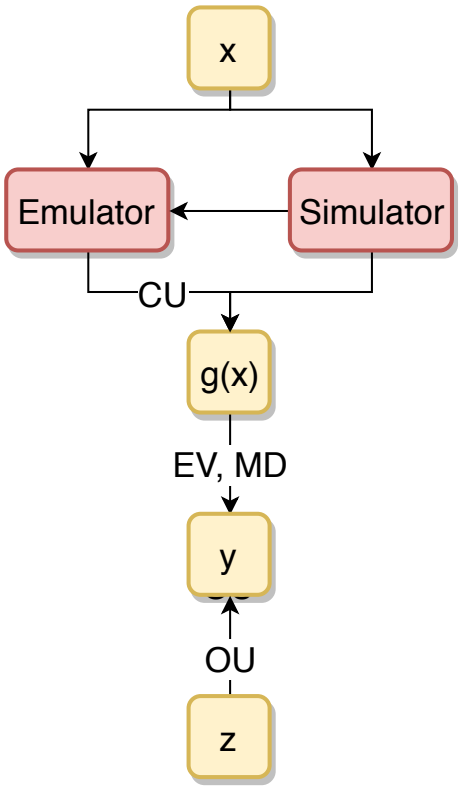


FIGURE 2.1: Uncertainty quantification along the HM process. Benchmark inputs $\mathbf{x}$ are evaluated by a simulator, where it causes MD and EV (if applied with stochastic simulators). Data obtained are used to train an emulator to reduce the running expense of a complex simulator, where it causes CU. Then model outputs are compared with the observation to find the corresponding fitting input domain, where it causes OU.

deterministic simulator as follows:

$$z = g(\mathbf{x}^*) \pm \sqrt{V_o} \pm \sqrt{V_m} \tag{2.2}$$

The above equation can be comprehended as follows: with inevitable measurement uncertainty V_o, one observation z is obtained from a physical process y, which is modeled by a simulator. Considering the simulation model itself is probably not adequate to describe y, the discrepancy error V_m is included. This sample is evaluated by the simulator, therefore avoids the code uncertainty V_c. The ensemble variability V_s is also unconcerned because this is a deterministic simulator.

2.3 History matching workflow

HM aims to find the input domain (representing values of engineering characteristics) that can provide a match between observations from a physical system and model outputs of the corresponding simulator. It iteratively cuts down the input space, leaving only the input domain with the potential to fit at each iteration.

The workflow of HM is the following:

1. To calibrate a simulator, a large number of samples and simulation runs are needed. Exhaustive exploration for a continuous input space is infeasible. Thus, HM starts from a sampling design to select benchmark samples for simulator evaluations.

2. Simulator evaluations are used to provide training data for an emulator.

3. An emulator makes inference about the simulator's output for unevaluated samples in order to reduce the running expense of a complex simulator.

4. An implausibility measure is used to compare model outputs with observed data in order to find out whether there exists a match. This implausibility threshold should allow diverse sources of uncertainty. At each iteration, the input domain considered implausible is ruled out.

5. HM stops when adequate non-implausible samples are obtained or the code uncertainty is small enough.

6. If the stopping criteria are not met, a new wave begins by resampling on the non-implausible domain. The need of evaluating the implausibility for each sample at each wave urges the research of a sampling method that can screen the input space. This dissertation proposes a reliability based strategy to solve the problem of sampling in the non-implausible domain.

7. Then steps 2 to 5 are repeated until HM finds the fitting input domain qualified for the stopping criteria.

The above workflow of HM is summarized in Figure 2.2. The iterative process of sam-

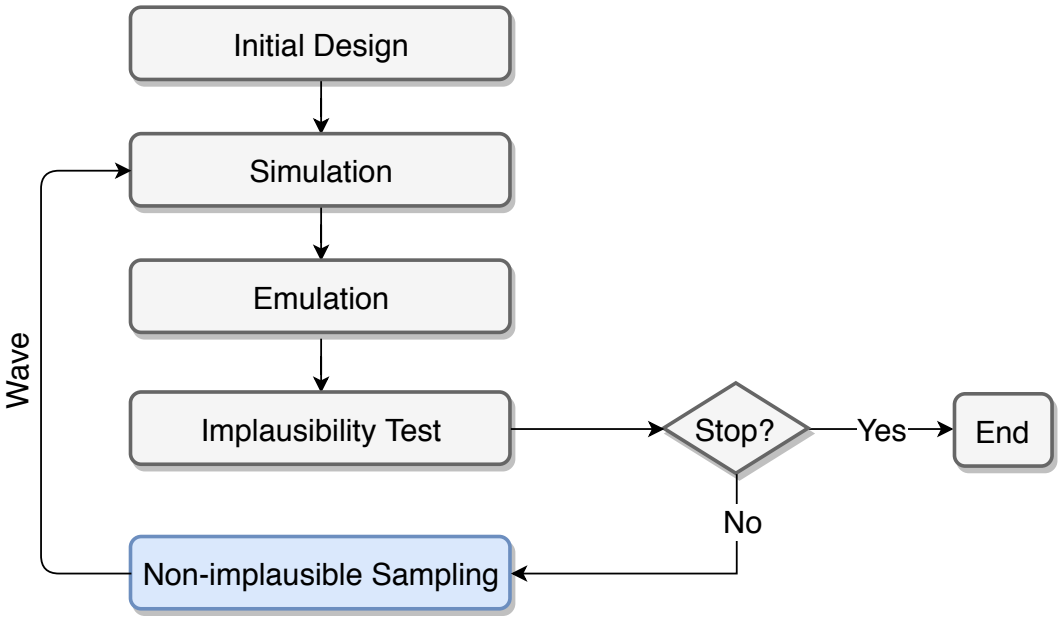


FIGURE 2.2: Typical workflow of HM. This work focuses on the sampling from the non-implausible domain efficiently.

pling on the non-implausible domain, evaluating more training data by the simulator, refining the emulator, and testing the implausibility is termed as a wave of refocusing [44]. The refocusing characteristic of HM leads to a higher concentration of non-implausible samples in every next wave. The shrinkage of the target space favors the use of an emulator because the model output is expected to behave smoothly on a reduced space. In the end, gradually precise waves will find the fitting input domain for high-dimensional space or complex simulators as long as there exists a match for the given data.

The components of the workflow are now explained in detail.

2.4 Initial design for simulation

An initial sampling design to select a set of benchmark samples to be evaluated by the simulator is the first step for a typical HM workflow. To calibrate the simulator, wide possible ranges of the input domain are considered at the beginning. The selection of benchmark samples is to run a design of experiments, which in this context is to decide which samples to run the model to most efficiently explore the whole input space, i.e. an

evenly spread of design points in design space and suitable projections. A set of samples with such property is called being space-filling.

2.4.1 Space-filling designs

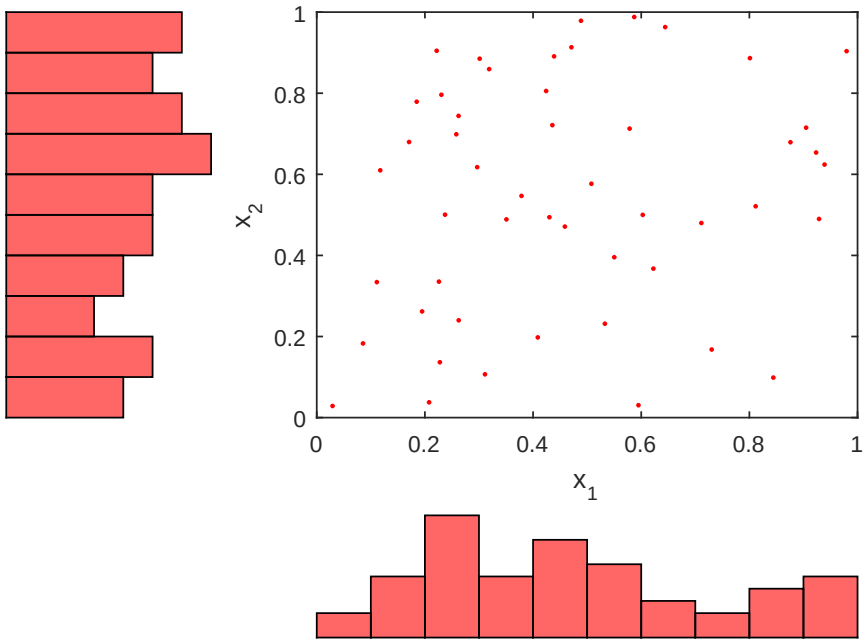
The Latin hypercube sampling (LHS) [59] is applied to efficiently explore the input space due to its good space-filling property. To obtain an LHS design, considering independent variables $\mathbf{x}_1, ..., \mathbf{x}_n$, first generate m samples separately from m regions divided by equal probability of the probability density function (PDF) for $\mathbf{x_1}$, then sample likewise for $\mathbf{x}_2$ to $\mathbf{x}_n$. Next permute randomly each variate's m sample points before assemble them together into an $m \times n$ matrix—the LHS design.

Because a good space-filling LHS should have samples evenly spread on every dimension, the corresponding projections should approximately follow uniform distributions. Therefore a Uniform histogram for each dimension can visualize the space-filling validity. Figure 2.3 shows the difference of a Uniform design and an LHS design with good space-filling property. One can see from this 2-D example that samples in an LHS design is more evenly spread.

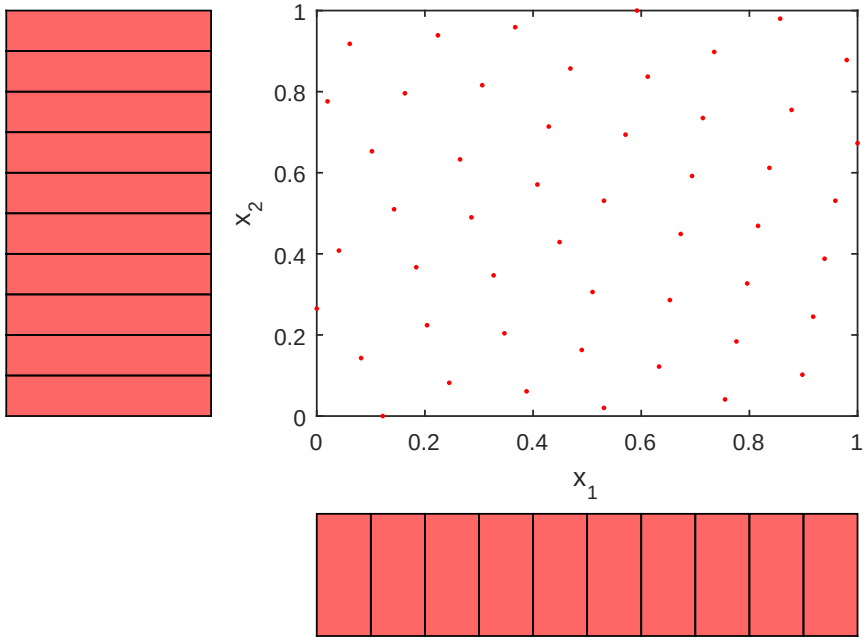
There exist several measures to quantify the space-filling quality, in order to choose a better LHS plan, which could more evenly cover the whole input space. Morris and Mitchell (1995) [60] defined a criterion function (Equation 2.3). Generally, the smaller the value of Φ_q, the better space-filling property of the sampling plan $\mathbf{x}$.

$$\Phi_q(\mathbf{x}) = \left(\sum_{j=1}^{m} J_j d_j^{-q} \right)^{1/q} \tag{2.3}$$

where q is an exponent coefficient, with recommended reasonable values $q = 1$, 2, 5, 10, 20, 50, and 100 [60] (the above six values of q should all be considered to calculated the smallest value of $\Phi_q(\mathbf{x})$ for a sampling plan $\mathbf{x}$), m is the total number of unique distance values for sample pairs and J is the multiplicity array, which stores the total number of



(a)Uniform design



(b)LHS design

FIGURE 2.3: 2-D samples of the same size of a Uniform design and an LHS design with good space-filling property, shown by its Uniform marginal histograms.

pairs separated by each unique distance value d_p, which could be calculated by:

$$d_p(\mathbf{x}_{i_1}, \mathbf{x}_{i_2}) = \left(\sum_{j=1}^{D} |\mathbf{x}_{j,i_1} - \mathbf{x}_{j,i_2}|^p \right)^{1/p} \tag{2.4}$$

where $\mathbf{x}_{i_1}$ and $\mathbf{x}_{i_2}$ are a pair of samples with distance d_p; D is the dimension of the input variables; p is the distance norm, and p is usually set to 1 in the form of the rectangular distance. Another common choice is $p = 2$ as Euclidean distance. There is no reasonable evidence in literature claiming that which one is better, therefore researchers could always choose the rectangular distance to have a less expensive computation [61]. To find an LHS plan which tends to minimize Φ_q, Forrester [61] proposed to use evolutionary algorithms (introduced by Box (1957) [62]). Evolutionary operation, just as its name implies, mutates a 'parent' many times to obtain 'offspring' and choose the best one with respect to a criterion as the next parent for the next generation. In the context of design of experiments, in each generation, samples of a parent sampling plan are slightly altered many times to form different sampling plans, among all sampling plans, the one yielding the smallest $\Phi_q(\mathbf{x})$ is chosen as the next parent. After many generations, the optimized sampling plan is obtained from random variations according to evolutionary survivals.

When it comes to cases of low dimensions and small sized samples, another criterion called the maximin definition (introduced by Johnson (1990) [63] then developed by Morris and Mitchell (1995) [60]) is more accurate for the quantification of space-filling. Among all sampling plans, a 'maximin' Latin hypercube has the largest value of the smallest distance between any two points, so that the samples are well spread. If the smallest distance is the same for some sampling plans, the second smallest distance is compared; if the second smallest distance is the same then the third smallest distance is compared, and so on in the same fashion.

2.4.2 Training data

Loeppky [64] provided a criterion for the number of training samples $n = 10 \times D$, where D is the input dimension. Once a sampling design is specified, the output $g(\cdot)$ is calculated by running simulations. For a stochastic simulation model, the simulator needs to run K times to estimate the mean value for each output $\hat{g}(\mathbf{x}_i)$ (K chosen wisely

by the calculation capacity of the computer):

$$\hat{g}(\mathbf{x}) = \frac{1}{K}\sum_{k=1}^{K} f_k(\mathbf{x}_i) \tag{2.5}$$

In both cases, the initial design for simulation or as to say the training data are obtained $\{(\mathbf{x_1}, g(\mathbf{x}_1)), ..., (\mathbf{x_n}, g(\mathbf{x}_n))\}$.

2.5 Bayesian emulator

2.5.1 Basic definitions

A Bayesian emulator [65–67], also called Gaussian process emulator or Kriging, is a stochastic approximation to the output of an expensive computer model, which is widely used as a surrogate for a complex simulator. Based on the simulation data, a Bayesian emulator interpolates the model output to reduce the running cost of an expensive simulator. This uses the Bayesian principle: for a fixed sample point, assign a prior probability distribution to the model output and update it with more observed data. It results in a predictive posterior distribution for the model output, which estimates not only the expectation but also the variance about the model output. The variance quantifies the uncertainty of an emulator's approximation. This property has made Bayesian emulators effective in multiple disciplines and applications, such as modeling for structural dynamic analyses [68, 69], stochastic mechanical responses [70], reliability assessments [71] among many others, where uncertainty quantification plays an important role.

A basic assumption in this dissertation is that the Bayesian emulator is of the form:

$$g(\mathbf{x}) = \sum_{i=1}^{p} h_i(\mathbf{x})^T \boldsymbol{\beta}_i + Z(\mathbf{x}) \tag{2.6}$$

where $g(\mathbf{x})$ is the emulator's output; $h_i(\mathbf{x})$ is a regression model, $i = 1, ..., p$; p is the dimension of the regression model; $\boldsymbol{\beta}_i$ is the regression coefficient; and $Z(\mathbf{x})$ is a zero-mean Gaussian process (see Definition 2.1). The function $h(\mathbf{x})^T\boldsymbol{\beta}$ models the global trend of the output, whereas Gaussian process $Z(\mathbf{x})$ models local variations. Common choices

for the regression model are $h(\mathbf{x})^T = [1]$, $h(\mathbf{x})^T = [1\,\mathbf{x}]$, $h(\mathbf{x})^T = [1\,\mathbf{x}\,\mathbf{x}^2]$ and $h(\mathbf{x})^T = [1\,\mathbf{x}\,\mathbf{x}^2\,\mathbf{x}^3]$. For example, $h(\mathbf{x})^T = [1]$ does not suggest any response of the output from the inputs' variation; $h(\mathbf{x})^T = [1\,\mathbf{x}]$ suggests linear response; $h(\mathbf{x})^T = [1\,\mathbf{x}\,\mathbf{x}^2]$ suggests quadratic response and so on. Note that, the emulator output interpolates the simulator output, for which $g(\cdot)$ is used to denote both.

Definition 2.1. *(Gaussian Stochastic Process)* $\mathbf{x} \in \mathbb{R}^D$, $GP(\cdot)$ is a Gaussian stochastic process if for any $D \geq 1$ and any choice from $\mathbf{x}_1, ..., \mathbf{x}_D$, the vector $[Z(\mathbf{x}_1), ..., Z(\mathbf{x}_D)]^T$ has a multivariate Normal distribution.

A zero-mean multivariate Normal distribution can be denoted as follows:

$$\begin{bmatrix} Z_1(\mathbf{x}) \\ . \\ . \\ . \\ Z_n(\mathbf{x}) \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ . \\ . \\ . \\ 0 \end{bmatrix}, \begin{bmatrix} k_{11} & \cdots & k_{1n} \\ . & & . \\ . & & . \\ . & & . \\ k_{n1} & \cdots & k_{nn} \end{bmatrix} \right) \tag{2.7}$$

where the variance-covariance matrix can be denoted as a constant variance σ^2 times the correlation matrix $\mathbf{A}$:

$$\sigma^2\mathbf{A} = \begin{bmatrix} \sigma^2 & \sigma^2 c(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \sigma^2 c(\mathbf{x}_1, \mathbf{x}_n) \\ \sigma^2 c(\mathbf{x}_2, \mathbf{x}_1) & \sigma^2 & & \vdots \\ \vdots & & \ddots & \\ \sigma^2 c(\mathbf{x}_n, \mathbf{x}_1) & \cdots & & \sigma^2 \end{bmatrix} \tag{2.8}$$

The correlation function $c(\mathbf{x}, \mathbf{x}')$, also called kernel, is designed to imply that if two samples are far from each other, the two outputs have a weak correlation, and vice versa. For a certain kernel, the smoothness coefficient θ governs the degree of correlation for a sample pair ($\mathbf{x}$ and $\mathbf{x}'$). Three common kernels are illustrated here. The exponential or the Gaussian correlation matrices (used by Figure 2.4) gives a smooth emulation output [2]:

$$c(\mathbf{x}, \mathbf{x}') = e^{-\frac{d^2}{\theta}} \tag{2.9}$$

The non-negative cubic correlation function gives a flexible cubic spline interpolant [26]:

$$c(\mathbf{x}, \mathbf{x}') = \begin{cases} 1 - 6(\frac{|d|}{\theta})^3 & |d| < \frac{\theta}{2} \\ 2(1 - \frac{|d|}{\theta})^3 & \frac{\theta}{2} \leq |d| < \theta \\ 0 & |d| \geq \theta \end{cases} \tag{2.10}$$

The $Matérn(5/2)$ correlation function [72] gives a moderate smoothing degree, which is in the form:

$$c(\mathbf{x}, \mathbf{x}') = \left(1 + \sqrt{5}\frac{|d|}{\theta} + \frac{5}{3}(\frac{d}{\theta})^2\right) e^{\frac{-\sqrt{5}d}{\theta}} \tag{2.11}$$

where d is the distance between $\mathbf{x}$ and $\mathbf{x}'$. A correlation function is chosen according to the knowledge of the simulator's behavior.

It can be seen, there are three unknown parameters from Gaussian process emulator that need to be estimated in order to obtain the emulation output: the smoothness parameter θ, the stationary variance σ^2, and the regression coefficient $\boldsymbol{\beta}$.

2.5.2 Emulator parameters

To estimate the parameters, the posterior distribution for the emulation output given the training data can be decomposed by the posterior distribution for the emulation output given both the parameters and the training data $\pi(g_0(\mathbf{x})|(\beta, \sigma, \theta), g(\mathbf{x}))$ and the posterior distribution of the parameters given the training data $\pi((\beta, \sigma, \theta)|g(\mathbf{x}))$, as following [73]:

$$\pi(g(\mathbf{x})|\Gamma(\mathbf{x})) = \int_{\boldsymbol{\beta}} \int_{\sigma} \int_{\theta} \pi(g_0(\mathbf{x})|(\boldsymbol{\beta}, \sigma, \theta), \Gamma(\mathbf{x}))\pi((\boldsymbol{\beta}, \sigma, \theta)|\Gamma(\mathbf{x}))d\theta d\sigma d\boldsymbol{\beta} \tag{2.12}$$

where $\Gamma(\mathbf{x})$ is the training data evaluated by the simulator $\Gamma(\mathbf{x}) = [g(\mathbf{x}_1), ..., g(\mathbf{x}_n)]^T$ and $g(\mathbf{x})$ is the emulation output.

Assuming the smoothness parameter θ involved in the correlation matrix A is known, the posterior distributions of $\boldsymbol{\beta}$ and σ can be obtained from the fact:

$$\pi(g(\mathbf{x})|\boldsymbol{\beta}, \sigma) \sim N(\mathbf{H}\boldsymbol{\beta}, \sigma^2\mathbf{A}) \tag{2.13}$$

Generally there's not much information about the behavior of the model output, hence non-informative priors for the parameters are preferred. Among non-informative priors, a Jeffreys prior ([74]) has a convenient feature that is invariant under reparameterization. No matter how to scale the parameters (e.g. changing to a logarithmic scale), it remains non-informative. It is proportional to the square root of the determinant of the Fisher information matrix [75]. In this case, the Jeffreys prior is proportional to σ^{-2}. Considering that σ^2 and $\boldsymbol{\beta}$ have a non-informative Jeffreys prior $p(\sigma^2, \boldsymbol{\beta}) \propto \sigma^{-2}$, based on Equation 2.13, the posterior distributions of β and σ are as follows:

$$\boldsymbol{\beta}|\sigma^2, \Gamma(\mathbf{x}) \sim N(\hat{\boldsymbol{\beta}}, \sigma^2(\mathbf{H}^T A^{-1}\mathbf{H})^{-1}) \tag{2.14}$$

$$\sigma^2|\Gamma(\mathbf{x}) \sim (n-p-2)\hat{\sigma}^2\chi^{-2}_{n-p} \tag{2.15}$$

where,

$$\hat{\boldsymbol{\beta}} = (\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{A}^{-1}\Gamma(\mathbf{x}) \tag{2.16}$$

$$\hat{\sigma}^2 = \frac{\Gamma(\mathbf{x})^T(\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{H}(\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H}^{-1})\mathbf{H}^T\mathbf{A}^{-1})\Gamma(\mathbf{x})}{n-p-2} \tag{2.17}$$

If Equation 2.13, Equation 2.16 and Equation 2.17 are combined, the posterior distribution for the output $g(\mathbf{x})$ follows the student t distribution:

$$\frac{g(\mathbf{x}) - E^*(g(\mathbf{x}))}{\hat{\sigma}\sqrt{v^*(\mathbf{x}, \mathbf{x}')}} \sim t_{n-p} \tag{2.18}$$

where $n-p$ is the degrees of freedom, $E^*(g(\mathbf{x}))$ is the desired expectation of the emulation output, $\hat{\sigma}\sqrt{v^*(\mathbf{x}, \mathbf{x}')}$ is the standard deviation of the emulation output, and:

$$E^*(g(\mathbf{x})) = h(\mathbf{x})^T\hat{\boldsymbol{\beta}} + t(\mathbf{x})^T\mathbf{A}^{-1}(\Gamma(\mathbf{x}) - \mathbf{H}\hat{\boldsymbol{\beta}}) \tag{2.19}$$

$$\begin{aligned} v^*(\mathbf{x}, \mathbf{x}') = \hat{\sigma}^2(c(\mathbf{x}, \mathbf{x}') - t(\mathbf{x})^T\mathbf{A}^{-1}t(\mathbf{x}') + (h(\mathbf{x}^T) - t(\mathbf{x})^T\mathbf{A}^{-1}\mathbf{H})(\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H})^{-1} \\ (h(\mathbf{x}')^T - t(\mathbf{x}')^T\mathbf{A}^{-1}\mathbf{H})^T) \end{aligned} \tag{2.20}$$

where, $t(\mathbf{x}) = [c(\mathbf{x}, \mathbf{x}_1), ..., c(\mathbf{x}, \mathbf{x}_n)]^T$.

The smoothness parameter θ in A can be estimated by cross validation [30]. The idea is as follows: omit one simulation sample point x_i, use the rest of the samples $x_{\sim i}$ to

build the emulator, then check the residual $d_i = E^*(g(\mathbf{x}_{\sim i})) - g(\mathbf{x}_{\sim i})$ between the emulator's output $E^*(g(\mathbf{x}_{\sim i}))$ and simulator's output $g(\mathbf{x}_{\sim i})$ at the omitted point. The same procedure is repeated until every simulation sample point is omitted once to obtain the sum of residuals $\sum_{i=1}^{n} d_i$ for a certain smoothness parameter. In the end, a smoothness parameter $\hat{\theta}$ that minimizes the sum of residuals $\sum_{i=1}^{n} d_i$ is chosen. Cross validation is a relatively expensive method. It is more suitable for low-dimensional applications. For large sized training data, the evolutionary algorithm [62] (explained in section 2.4 for searching a space-filling LHS design) can be used to search for a $\hat{\theta}$ maximizing the likelihood $\mathcal{L}(g(\mathbf{x_1}), ..., g(\mathbf{x_n})|\theta)$.

2.5.3 A 1-D example using a Gaussian process emulator

A simple example is provided to give a basic idea, that is using a Gaussian process emulator to make interpolation about a deterministic one-dimensional function: $g(x) = 5 + x + cos(x)$. Five benchmark samples are evaluated using the simulator (the function) in order to obtain the training data $(-4.334, 0.2966)$, $(-2.054, 2.4814)$, $(0, 6.0000)$, $(2.054, 6.5894)$, $(4.334, 8.9646)$ for the Bayesian emulator. The result of the Bayesian emulator is a predictive posterior distribution for the model output at each sample point, shown by Figure 2.4.

2.5.4 Validation

There are some reasons when an emulator might fail to capture the patterns of a simulator:

1. Inappropriate choice of polynomials;
2. Inappropriate choice of correlation functions;
3. Inappropriate choice of design of experiment;
4. Inappropriate estimation of parameters.

Therefore, an emulator must be validated before use. Cross validation [76] has a convenient feature that is no need of new training samples. The principle is similar as the one

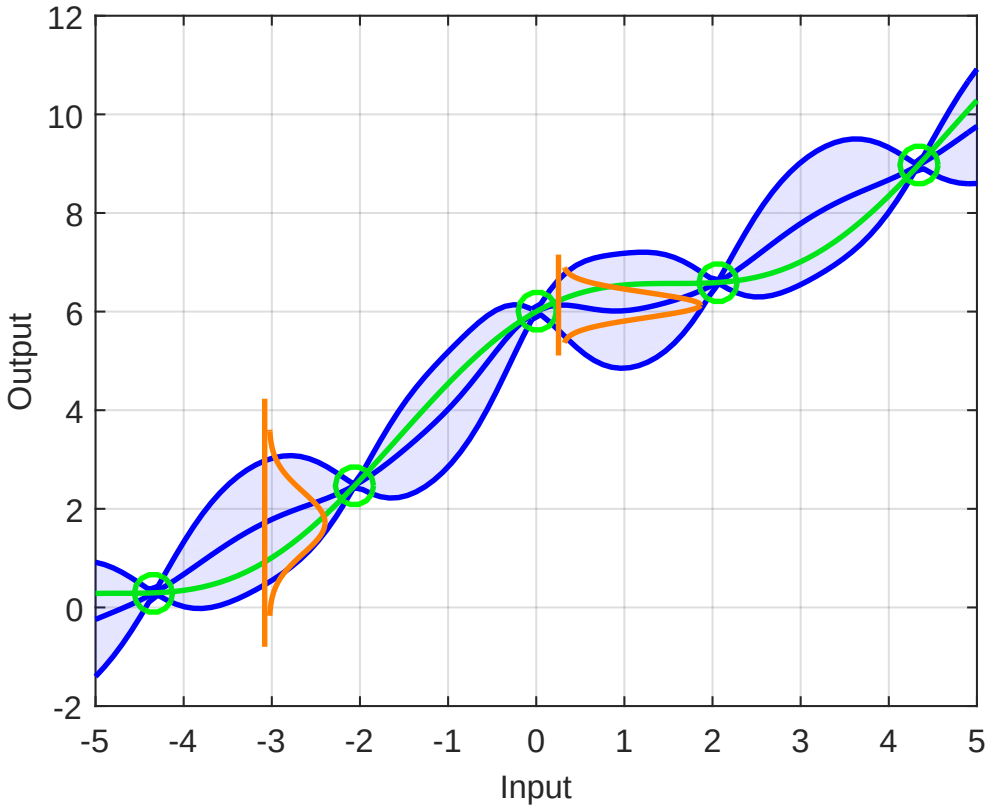


FIGURE 2.4: Emulation of a simulator. The green circles are the training data. The predicted mean of the emulator is denoted by the dark blue line in the middle, enveloped by 95% credible intervals (two folded standard deviation of the estimated model output) denoted by the shaded color. For comparison, the original simulation output on the input space [-5,5] is presented by a green line. The orange lines demonstrate the posterior probability distributions of the predicted output at input points -3 and 0.25.

used for estimating the smoothness coefficient. At each omitted simulation sample point, if the emulation output is close to the simulation output, this emulator is accepted.

There also exists a Mahalanobis distance diagnostic [77]. Mahalanobis distance [78] is a method to detect outliers in a multi-dimensional space, even though the outlier cannot be detected in any projections. For example, shown by Equation 2.22, the red outlier cannot be detected from its any 1-D projections.

Such outliers are detected by testing a sample's Mahalanobis distance MD deviated from the mean considering the covariance of the samples.

$$MD = \sqrt{(x_i - \mu)^T \Sigma^{-1} (x_i - \mu)} \quad i = 1, ..., k \tag{2.21}$$

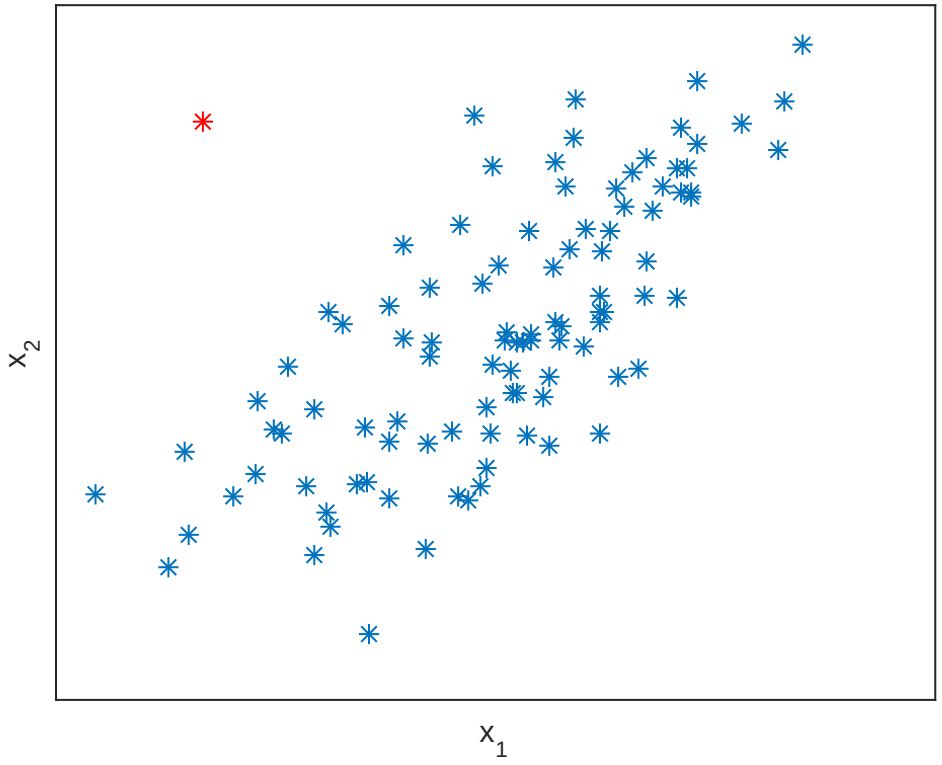


FIGURE 2.5: An illustration of an outlier (in red), which cannot be detected in each independent dimension.

where, x_i are the samples under tested, μ is the mean and Σ is the covariance matrix of x_i.

In the case of validating an emulator, more data x_i, $i = 1, ..., k$ are evaluated by the simulator and if their Mahalanobis distances are far deviated from the mean of the model output estimated by the emulator, that means the emulator is not acceptable. Bastos and O'Hagan [77] judge this using an auxiliary variable M:

$$M = (g(\mathbf{x}_i) - E^*(g(\mathbf{x})))^T \Sigma (g(\mathbf{x}_i) - E^*(g(\mathbf{x}))) \tag{2.22}$$

where Σ is the covariance matrix of the new training data $\mathbf{x_i}$. For acceptable emulators, M should follow a F-Snedecor distribution with parameters $n/10$ and $n-p$ and the calculated value of M should be close to the mean value $n/10$ of the reference F-Snedecor distribution. Otherwise it means that there exist large deviations between some estimations of the emulator and the model output, so that the emulator is unacceptable.

To improve an unacceptable emulator, the following methods can be tried:

1. Rebuild the emulator with more data, trying to improve the estimation of parameters (β, θ, σ).

2. Change the assumed prior or correlation function. For example, because the traditional exponential correlation matrix [79] makes the emulation output too smooth in many practical cases [80], the $Matérn(5/2)$ correlation matrix with a moderate smoothing degree should be considered. In the case there is no knowledge about the simulation model, a constant polynomial is preferred [30], without assuming any form of output response.

2.6 Implausibility measure

HM defines an implausibility measure to identify to what extend a model output matches the observation. According to this implausibility measure, a certain threshold decides whether the corresponding model input is non-implausible (retained) or implausible (discarded). In order to avoid excluding possible fitting samples due to the experimental error, inadequacy of the model, the emulator's approximation and other unexpected sources of uncertainty, the implausibility measure should consider all aforementioned uncertainties in HM and the implausibility threshold should allow some extra uncertainty margins.

Let $I(\mathbf{x})$ denote the implausible measure [48] that an output matches the specified observation, quantified via the difference between them with the consideration of the uncertainties in HM:

$$I(\mathbf{x}) = \frac{|z - E^*(g(\mathbf{x}))|}{\sqrt{V(\mathbf{x})}} \tag{2.23}$$

If at a sample point $\mathbf{x}$, the implausibility measure $I(\mathbf{x})$ returns a small value, it is very likely that this input can provide an acceptable match between the model output and the experimental data.

For a fixed sample point x, the implausibility measure can be considered as a variable with a continuous unimodal distribution. To decide the threshold, Vernon et al. [44]

suggest to follow the Pukelsheim three sigma rule [81], which implies that for a variable (even with a highly skew or heavy tailed distribution), around 95%, or as to say nearly all values in its probability distribution lie within a three folded standard deviation band around the mean. A threshold of $I(\mathbf{x}) \leq 3$ then implies that as long as a model input x makes its implausibility measure belong to the distribution with the mean equal to the observation $E^*(g(\mathbf{x}))$ and the variance equal to the uncertainties $V(\mathbf{x})$, the input is retained. The threshold reflects how close the non-implausible samples' output are to the observation. It is possible to tighten the threshold to smaller values. For example, instead of 3, if 1 is applied, only samples with their output lying within the domain around the observation with one standard deviation are retained. In chapter 5, the resulting inputs using threshold 1 will be compared with using threshold 3.

If multiple independent physical quantities are considered, an input domain is said non-implausible if it can provide matches for all outputs $g_j(\mathbf{x})$ $j = 1, ..., r$ at the same time:

$$\frac{|z_1 - E^*(g_1(\mathbf{x}))|}{\sqrt{V_1(\mathbf{x})}} \leq 3, \frac{|z_2 - E^*(g_2(\mathbf{x}))|}{\sqrt{V_2(\mathbf{x})}} \leq 3, ..., \frac{|z_r - E^*(g_r(\mathbf{x}))|}{\sqrt{V_r(\mathbf{x})}} \leq 3 \tag{2.24}$$

Hence, the implausibility threshold for multiple outputs is as follows:

$$I(\mathbf{x}) = max(I_j(\mathbf{x})) \leq 3, \ j = 1, ..., r \tag{2.25}$$

If for a particular physical quantity, multiple observations are available e.g. z_j and z'_j, then an input domain is said non-implausible not only when satisfying $\frac{|z_j - E^*(g_j(\mathbf{x}))|}{V_j(\mathbf{x})} \leq 3$ but also when satisfying $\frac{|z'_j - E^*(g_j(\mathbf{x}))|}{V_j(\mathbf{x})} \leq 3$. Hence, the implausibility threshold for multiple observations is as follows:

$$I(\mathbf{x}) = min(I_j(\mathbf{x})) \leq 3 \tag{2.26}$$

Sample points that fail the implausibility threshold are considered implausible, since they hardly show the potential to provide a match. Non-implausible samples defined by $I(\mathbf{x}) \leq 3$ are not ruled out given the current information, but with possibility to be ruled

out in sequential waves. A new wave of HM begins by sampling from the non-implausible domain to refocus only on the potential fitting input domain.

2.7 Sampling design for new waves

Initial simulation points (benchmark samples evaluated by the simulator to train the emulator at the first wave) and initial emulation points (interpolating samples evaluated by the emulator to explore more of the input space at the first wave) can both be generated by space-filling LHS plans. For sequential waves, although one of the advantages of HM is to only refocus on the non-implausible domain to find the fitting inputs from a high-dimensional space, to screen the input space—sampling only from a non-implausible domain can be challenging. The non-implausible domain may be disconnected, orders of magnitude smaller than the original space, or with a nontrivial topology. Different sampling schemes are introduced for generating new simulation points and emulation points to solve this problem.

2.7.1 Sampling new simulation points

For each new wave, HM feeds new training data to the emulator to make it more accurate. The variance of the emulator's output is expected to be reduced so that more implausible domain can be detected and the non-implausible input space can be further cut down. Some sampling methods update solely in the region of interest to specifically enhance the local accuracy, and other sampling methods update throughout the design region to enhance the general accuracy.

Selecting training data for an emulator has been an active challenge in the field of optimization. Inspired by Jones [82], Forrester et al. [61] pointed out a local exploitation, a global exploration, and combination methods of exploration and exploitation when seeking optimum values using emulators. The prediction based local exploitation converges on the global optimum by iteratively adding a point at the current prediction of the local optimum. The error based global exploration iteratively uses the point with the maximum predicted error as the updating point, taking advantage of the Gaussian

emulator permitting the calculation of the standard deviation of the predicted output. The balanced exploitation and exploration method infills points that have the largest 'probability of improvement' (the measure is defined in [61]). The conditional likelihood approach selects the optimum by assuming that the emulator passes through imputed optimum points. Thus intuitively, in the case of HM, training data can be added in the domain where the variance of the emulator output is large, or in the domain with a large probability to be non-implausible, which is close to the current non-implausible domain, or on the whole input space to improve the global accuracy.

For HM, it is inefficient to add new samples extensively from the whole input space to enhance the global accuracy, while the implausibility threshold of HM already traces down the most potential fitting input space—the non-implausible input domain. Therefore on the purpose of embedding the method of selecting training data with HM, the infill points are only selected from the non-implausible samples identified by the current wave. Using the existing non-implausible samples has a key advantage. It avoids sampling the whole input space and evaluating each sample's implausibility to select the ones belong to the 'refocusing' domain. Considering the non-implausible domain may be disconnected and the emulator needs to be refined on each of the non-implausible region, the Density Based Spatial Clustering of Applications with Noise (DBSCAN) [53]—a widespread clustering algorithm—is applied for sampling new simulation points. It detects disconnected non-implausible domain, then new training samples are selected from each region trying to evenly cover the disconnected non-implausible domain. In the end, the simulator runs on these new samples to obtaining more training data. Since this chapter only trying to explain every components of HM, the implementation of this proposed method will be discussed in detail at section 4.6.

2.7.2 Sampling new emulation points

At each wave, based on the current non-implausible domain, new emulation samples should be generated in order to find more non-implausible samples and further cut down the implausible domain. To target the current non-implausible domain, the implausibility measure has to be evaluated for each sample at each wave. This brings

computation difficulties. A high dimensional input space may require a great number of samples to find only a few samples on the current non-implausible domain. Thus an efficient sampling method to screen the input space is desired.

The most intuitive sampling approach would be to generate samples well spread on the whole input space, then discard implausible ones that fail the implausibility threshold. This acceptance-rejection strategy is time-consuming and inefficient, especially when the fitting input domain is only a small fraction of the original input domain. Due to this, other sampling approaches for multi-wave computer experiments (that involve iterative processes and need resampling for each iteration, such as HM) have been developed in the literature. Williamson and Vernon [47] proposed an implausibility driven evolutionary Monte Carlo algorithm (IDEMC). It provides uniform designs for the target space using an implausibility ladder derived from temperature based MCMC algorithms [83]. Evolutionary Monte Carlo [84] are applied to generate new samples w.r.t. the ladder iteratively. However, amongst the complexities, IDEMC presents the difficulty of choosing of an appropriate implausibility ladder. Andrianakis et al. [48] suggested a perturbation approach that generates normally distributed samples with mean value set on each point from the non-implausible domain of the current iteration. The variance-covariance matrix of the non-implausible samples is scaled to make a relatively flat distribution. In this way, only 20% of the samples remain in the non- implausible domain, which makes adequately different samples. But for this perturbation method, it is difficult to tune an appropriate inflation coefficient.

Given that the non-implausible domain can be far smaller than the original input domain, it can be interpreted as a failure set in the engineering reliability context, where sampling is done through rare event simulation. In this reason, Subset Simulation [52]—a widely used technique in engineering reliability computations—is proposed to solve the sampling problem for HM, which will be discussed in detail at chapter 3.

2.8 Stopping criteria

HM stops when the purpose of model calibration is accomplished. Three common stopping criteria exist in literature:

1. Cumming and Goldstein [85] solved the problem of finding the suitable values of permeability and porosity from different regions of oil wells to match a specified oil production. Their stopping criterion was that the computing budget (mainly in terms of time) is exhausted.

2. Andrianakis et al. [48] made a comparison between a newly established cosmologic model and the real observations contributing to the evolution of galaxies. Their stopping criterion was that more than 50% non-implausible samples are found in a wave.

3. Vernon et al. [44] found the partnership formation to match the demographic, behavioral, and epidemiologic characteristics of HIV prevalence. Their stopping criterion was that the CU is smaller than the rest sources of uncertainty (OU, EV and MD).

2.9 Summary

In this chapter, HM—a model calibration method suitable for high-dimensional input space and complex simulators—was introduced. Important components of HM have been separately explained. The iterative nature enables HM to focus on the non-implausible input domain with a potential to provide a match between observations and model outputs, while the input domain with low plausibility is discarded. However this powerful refocusing technique brings sampling challenges. The implausibility measure has to be applied in turn to determine whether each input belong to the non-implausible domain or the implausible domain. Thus a sampling method that can fastly screen the inputs according to their implausibility measure is desired. In this dissertation, this problem is

solved by using a reliability based sampling technique. The detailed solution is presented in following chapters.

Chapter 3

Reliability Analysis and Subset Simulation

3.1 Introduction

Engineering systems are characterized by a series of individual components governed by a large number of safety laws, such as massive, connected and multilevel power grids. For safety reasons and proper functioning, they should always be designed to have a low failure probability. This comes to engineering reliability analysis, which aims at quantifying the probability of failure for engineering systems [86]. By definition, reliability is complementary to probability of failure [87]:

$$Reliability = 1 - Failure\ Probability \tag{3.1}$$

Despite the variability of engineering problems, a failure event F can always be represented as that a system's response value y exceeds the specified critical value y^*, denoted by $F = y > y^*$. The response variable y often depends on a set of 'input variables' $x_1, x_2, ..., x_d$ as well as their performance function: $g(\cdot)$ [87], denoted by:

$$y = g(x_1, x_2, ..., x_d) \tag{3.2}$$

Due to the complexity of modern engineering systems, and the inherent uncertainties of this complexity, the input variables can be modelled probabilistically, that is, they are stochastic. For example, when evaluating the wind velocity pressure w_0, it is always affected by local weather conditions. It has different probabilities to be a set of possible values. The failure probability of the response variable is assessed by quantifying the input variables' uncertainty propagated through the performance function. The performance function can be an analytical formula, an empirical formula, or a complex simulator such as a finite element model.. The joint probability distribution of input variables can be specified by analysts, but a complex performance function can make the probability distribution of the resulting response variable intractable. That is the value of the response variable can be calculated by the set of given input variables, but no other information is available, such as the desired response variable's probability distribution. For example, a finite element model [88] subdivides a large system into hundreds of thousands of smaller and simpler elements to analyze, then equations applied on these finite elements are assembled into a global system of equations, where mathematically propagating the PDFs of the random input variables is difficult [89–91].

Now that the probability distribution of the response variable is unknown, evaluating a system's failure probability needs to know when the failure occurs, which can be represented by a set of failure scenarios $x \in F$. The probability of failure, denoted by p_F, can then be calculated by the probability integral of each failure scenario $x \in F$ [87]:

$$p_F = P(y > y^*) = P(\mathbf{x} \in F) = \int_F \eta(x)dx \tag{3.3}$$

where $\eta(x)$ is the joint PDF of input variables and F is the failure region of a subset in the input space, where failures occur with certain configurations of input values.

Estimating this multidimensional integral leads to the usage of the Direct Monte Carlo method [92, 93]. It estimates the failure probability as the proportion of failure samples out of all samples of the system's behavior. The Direct Monte Carlo algorithm counts failure samples scanned from all kinds of possible behaviors in the system of interest [94]

and

$$p_F \approx \frac{1}{m} \sum_{i=1}^{m} \mathcal{I}_F(\mathbf{x}) \tag{3.4}$$

where m is the number of all samples and $\mathcal{I}_F$ is an indicator function that counts the number of failure scenarios:

$$\mathcal{I}_F(\mathbf{x}) = \begin{cases} 1, if\ \mathbf{x} \in F, \\ 0, if\ \mathbf{x} \notin F. \end{cases} \tag{3.5}$$

A high reliability represents a proper designed engineering system, which means the corresponding failure probability is often in the order of $10^{-3} \sim 10^{-6}$ [87]. Obtaining failure samples can be difficult in the case of e.g. $p_F = 10^{-6}$, ideally one failure sample when sampling 10^6 samples. Sampling techniques trying to identify the important region that gives more information about finding the failure region have been developed, such as the First-Order Reliability Method and the Second-Order Reliability Method [95–97], Importance Sampling [98], and Subset Simulation [52] among others. Among these more 'advanced' sampling techniques, Subset Simulation does not make any assumption about the shape of the failure surface (linear or quadratic, etc.), while others rely on a prudent choice of the importance sampling density [99–102]. This means Subset Simulation does not require knowledge of the system in the failure region [52, 103]. Hence, this work considers to use Subset Simulation to solve the problem of sampling on the non-implausible domain for HM.

3.2 Subset Simulation

A failure set is small for a reliable engineering system thus sampling from it is challenging. Subset Simulation [52] is an advanced Monte Carlo method that efficiently targets small failure sets (with $p_F \leq 10^{-3}$) and computes rare failure probabilities for engineering reliability problems, especially for high-dimensional systems that a failure set can be orders of magnitude smaller than the input space [104]. Subset Simulation significantly reduces the needed number of samples based on a simple idea that is to decompose a rare event of small probability into sequential conditional events that are more likely to

happen and easier to sample from [87]. The conditional/intermediate events are a series of nested subsets $F_k, F_{k-1}, ..., F_1$, satisfying $F = \{\mathbf{x} : g(\mathbf{x}) < y^*\} = F_k \subset F_{k-1} \subset \cdots \subset F_1$. Thus, the probability (of failure) for the target event F can be calculated by the following equation according to the product rule in Probability Theory [105, 106]:

$$\begin{aligned} P(F) = P(F_k) &= P(F_1)\frac{P(F_2)}{P(F_1)}\frac{P(F_3)}{P(F_2)}\cdots\frac{P(F_{k-1})}{P(F_{k-2})}\frac{P(F_k)}{P(F_{k-1})} \\ &= P(F_1)P(F_2|F_1)...P(F_k|F_{k-1}) \end{aligned} \tag{3.6}$$

where $P(F_i) = P(F_iF_{i-1}) = P(F_i|F_{i-1})P(F_{i-1})$ since $F_i \in F_{i-1}$, $i = 2, ..., k$.

The algorithm to decompose intermediate events F_1 to F_{k-1} towards the failure event F is as follows. Every intermediate event is termed a 'level' in Subset Simulation. At the initial 1^{st} level, the known information is the input variables and the performance function. A certain number of samples can be randomly generated by the inputs' joint PDF to obtain the responding output (via the performance function). A subset of these samples with their output values more close to the critical threshold are selected as seeds to propagate samples for the next level. The conditional sampling based on seeds to propagate samples is achieved by Markov Chain Monte Carlo (MCMC) [83] (see section 3.3). Each seed generates a sequence of conditional samples, which forms a 'Markov chain' of samples. Repeating the process of selecting seeds and generating conditional samples constructs intermediate levels and eventually the target event. In the end, given the intermediate samples, failure samples can be found without a large sample size.

The most important aspects of implementing the algorithm are to define thresholds for intermediate levels and to decide the number of seeds. In each level, let n_L be the number of samples, and let p_L be the 'level probability', such that the number of seeds can be denoted by $n_s = n_L \times p_L$, and the number of conditional samples generated by MCMC can be donated by $n_c = (1-p_L)/p_L$. Let Y_{SS_i} denote the sequence of thresholds for intermediate levels, such that according to Equation 3.6 the failure probability can also be expressed by:

$$P(y > y^*) = P(y_1 > Y_{SS_1})P(y_2 > Y_{SS_2}|y_1 > Y_{SS_1})...P(y_k > Y_{SS_k}|y_{k-1} > Y_{SS_{k-1}}) \tag{3.7}$$

To calculate $P(y > y^*)$, one needs to estimate $P(y_1 > Y_{SS_1})$ and $P(y_i > Y_{SS_i}|y_{i-1} > Y_{SS_{i-1}})$, $i = 2, ..., k$. If probabilities of all intermediate events $P(y_1 > Y_{SS_1}) = P(y_i > Y_{SS_i}|y_{i-1} > Y_{SS_{i-1}})$, $i = 2, ..., k-1$ are assigned to be equal to the level probability p_L, the failure probability can be computed as:

$$P(y > y^*) = p_L^k \cdot P(y_k > Y_{SS_k}|y_{k-1} > Y_{SS_{k-1}}) \tag{3.8}$$

The intermediate thresholds for F_i and F_{i-1} are calculated as:

$$Y_{SS_i} = \frac{g_i(x_{n_s}) + g_i(x_{n_s+1})}{2} \tag{3.9}$$

where $g_i(x_{n_s})$ is the n_s largest response value of the seeds for level i. One can see that in this way, exact $n_s = n_L \times p_L$ seeds are qualified with p_L gives a direct estimate of the probabilities for intermediate levels.

The level probability p_L and the number of samples in each level n_L are considered as two fundamental parameters for Subset Simulation. The values for them are determined by the user in principles of:

1. The level probability p_L and the number of samples in each level n_L must ensure integer values for both the number of seeds $n_s = n_L \times p_L$ and the number of the conditional samples generated by each seed $n_c = (1 - p_L)/p_L$;

2. The level probability should satisfy $p_L \leq 0.5$, so that at least one sample can be generated based on each seed.

3. A conventional choice in practice for the level probability is $p_L = 0.1$ [87], which strikes a balance for the intermediate thresholds such that the conditional probability $P(y_i > Y_{SS_i}|y_{i-1} > Y_{SS_{i-1}})$ is not very large, resulting in inefficient too many intermediate levels, as well as it is not very small, resulting in an unwanted rare event problem for the conditional sampling;

4. The number of samples for each level n_L should strike a balance between the computation speed and the coverage of the input space.

The decomposition process repeats until at a level the failure samples can be generated as easy as the conditional samples for each intermediate event:

$$n_F = \frac{\sum_{i=1}^{n_L} \mathcal{I}_F(\mathbf{x})}{n_L} > p_L \tag{3.10}$$

where n_F is the proportion of failure samples out of the whole samples in a level and $\mathcal{I}_F(\cdot)$ is an indicator function that counts the number of samples in the failure set. Equation 3.10 also gives a direct estimate of the failure probability for the last level $P(y_k > Y_{SS_k} | y_{k-1} > Y_{SS_{k-1}})$. Thus, according to Equation 3.8, the desired failure probability can eventually be estimated by:

$$\hat{P}(y > y^*) = p_L^k \frac{\sum_{i=1}^{n_L} \mathcal{I}_F(\mathbf{x})}{n_L} \tag{3.11}$$

Figure 3.1 illustrates the decomposition process of Subset Simulation for any two dimensions of the input variables x_i, x_j, $i, j = 1, ..., D$. Figure 3.2 summarizes the corresponding workflow.

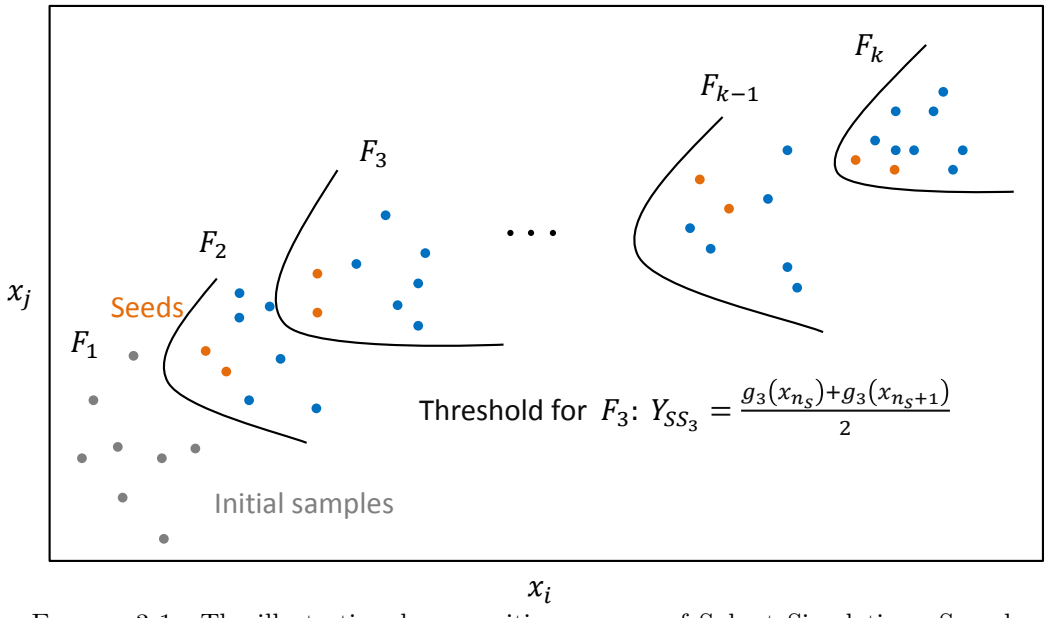


FIGURE 3.1: The illustrative decomposition process of Subset Simulation. Samples of intermediate conditional levels are denoted by blue dots. The orange dots denote the samples more close to the critical value in a level, which are used as seeds to generate more samples. The original input space has samples in gray. The intermediate thresholds are represented by black curves, e.g. the value of the threshold for for level 3 is labeled at the figure.

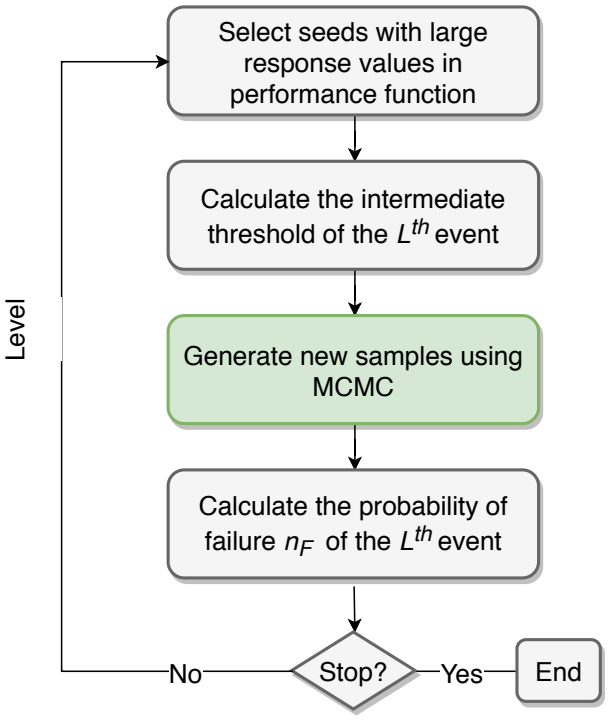


FIGURE 3.2: The workflow for Subset Simulation sampling for reliability problems. The green box emphasizes the step of MCMC sampling to be investigated in the following section.

3.3 MCMC schemes for Subset Simulation

The conditional sampling algorithm is the heart for Subset Simulation to generate conditional samples propagating towards the target rare event, which is performed by MCMC. MCMC represents a class of conditional sampling techniques [107, 108]. An investigation was made into different MCMC sampling schemes to determine suitable ones for HM.

3.3.1 Modified Metropolis algorithm

The original Subset Simulation algorithm presented by Au and Beck [52] uses the component-wise Metropolis-Hastings algorithm [109, 110], known as Modified Metropolis algorithm (MMA), to generate conditional samples that belong to intermediate less rare events. As one of the MCMC conditional sampling schemes, MMA generates sample candidates using proposal PDFs with their mean values based on the current samples.

Sample candidates with their respond values more close to the critical value are more likely to be accepted. MMA confirms to accept or reject this candidate depending on whether the sample belongs to the current intermediate event, which is determined by if the sample's respond value is larger than the corresponding intermediate threshold.

MMA is derived from the Metropolis-Hastings algorithm. Considering that if sample candidates often get rejected and stay at the same coordinates, they cannot populate the target space, hence MMA tries to make the samples move. Considering that input variables $x_1, ..., x_d$ are assumed mutually independent, they can be generated dimension-wise. In this way, even though the movements along some dimensions get rejected, the movements along other dimensions may be accepted. The implementing steps of MMA for intermediate event F_i is as follows:

Modified Metropolis algorithm

1. Propose a random standard Normal move $\alpha \sim N(\mathbf{x}_k, 1)$ in each dimension from each seed $\mathbf{x}_1, ..., \mathbf{x}_k, ..., \mathbf{x}_{n_s}$, or alternatively adaptive MMA: a Normal move using the variance of the current sampling seeds $\mathbf{x}_1, ..., \mathbf{x}_{n_s}$;

2. Accept the candidate movement $\xi_1 = \alpha$ with probability $min\{1, \frac{\phi(\alpha)}{\phi(\mathbf{x}_k)}\}$, otherwise stay at $\mathbf{x}_k$, $\xi_1 = \mathbf{x}_k$ (the first rejection event);

3. Remain $\mathbf{x}_k$ if the movement does not belong to the intermediate event F_i $\xi_1 \notin F_i$ (the second rejection event);

4. Sequentially conduct the conditional sampling, obtaining ξ_3 based on ξ_2, obtaining ξ_4 based on ξ_3, ..., until obtaining ξ_{n_c} based on ξ_{n_c-1}.

Au and Wang [87] pointed out two important aspects for MMA. First, in an adaptive manner, using the standard deviation of the current samples as the standard deviation of the proposal distribution balances the samples' acceptance rate and diffusion rate. If the standard deviation of the proposal distribution is very large, candidates are far from the current samples (that are known with larger response values) hence are more likely to have lower response values and get rejected. If the standard deviation of the proposal distribution is very small, candidates are too close to the current samples, resulting

in that they cannot 'freely' explore the input space, e.g. in the case where the input distributions have two modes, samples may be trapped locally—only moving around one of the modes. Second, the efficiency of MCMC is insensitive regarding to the form of the proposal PDF, therefore a symmetric PDF capable of covering the whole input space would be sufficient. Figure 3.3 shows the working mechanism of the MMA algorithm.

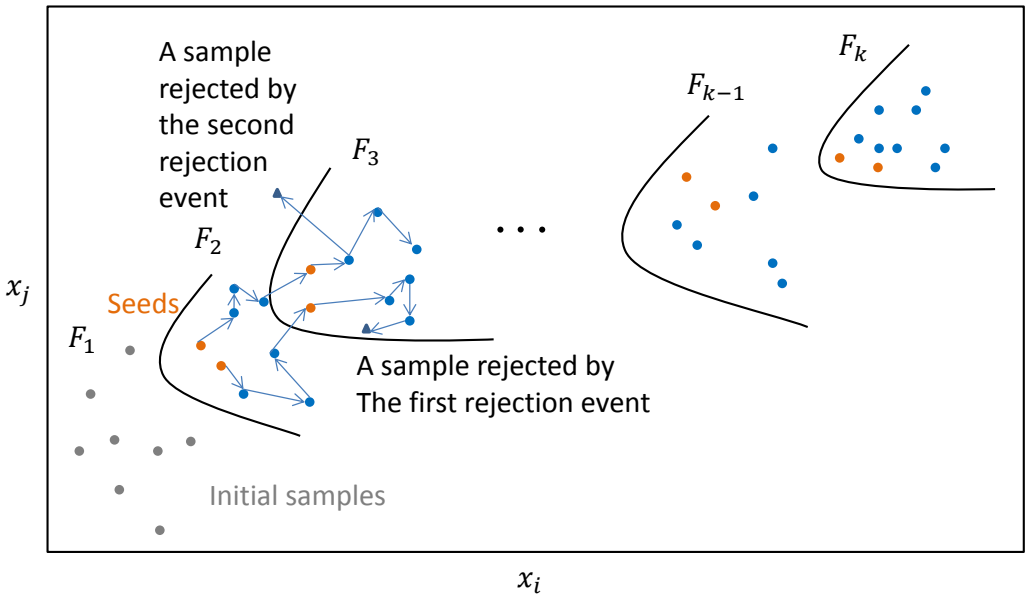


FIGURE 3.3: Samples generated by MMA at intermediate conditional levels are denoted by blue dots. The orange dots denote the samples more close to the critical value in a level, which are used as seeds to generate more samples. Rejected samples are denoted by purple dots. The original input space has samples in gray.

For convenience of computation, many reliability methods, including Subset Simulation, assume i.i.d. Gaussian input variables, including Subset Simulation [111]. This is not a limitation since dependent variables start from independent variables, as well as input variables of non-Gaussian distribution can be transformed to Gaussian distribution. The universal transformation rule for any continuous distribution being transformed to a Gaussian distribution is termed as Gaussianization [112]. It can be seen that for univariate distributions, the Gaussianization process is as follows:

$$\mathbf{x}_{t_i} = \Phi^{-1}(F(\mathbf{x}_i)) \tag{3.12}$$

where $F(\cdot)$ is the original CDF for an input variable. Every point in the new standard Normal distribution has the same cumulative probability as in the original distribution.

The likelihood function (omitting the subscript t for simplicity) for the transformed Normal distribution is $\mathcal{L}(\mathbf{x}_1, ..., \mathbf{x}_n) = \prod_{k=1}^{n} \Phi(\mathbf{x}_k)$.

The MMA and (adaptive MMA) are not the only sampling schemes available in the literature [113]. Some alternative MCMC sampling schemes are emerged trying to increase the number of acceptable samples. They are outlined in following sections. Their advantages and disadvantages are discussed.

3.3.2 Repeated sample generation

Trying to increase the acceptable samples, Santoso et al. [114] proposed to generate more sample candidates. They repeat the sample generation process until one candidate is not rejected by the first rejection event. The repeated sample generation approach for sampling in the i^{th} intermediate event F_i can be described as follows:

Repeated sample generation

1. Propose a random standard Normal move $\alpha \sim N(\mathbf{x}_k, 1)$ in each dimension from each seed $\mathbf{x}_1, ..., \mathbf{x}_k, ..., \mathbf{x}_{n_s}$, or alternatively in an adaptive manner: a Normal move using the variance of the current sampling seeds $\mathbf{x}_1, ..., \mathbf{x}_{n_s}$;

2. Accept the candidate movement $\xi_1 = \alpha$ with probability $min\{1, \frac{\phi(\alpha)}{\phi(\mathbf{x}_k)}\}$, otherwise go back to step 1;

3. Remain at $\mathbf{x}_k$ if the movement does not belong to the intermediate level F_i $\xi_1 \notin F_i$;

4. Sequentially conduct the conditional sampling: obtaining ξ_3 based on ξ_2, obtaining ξ_4 based on ξ_3, ..., until obtaining ξ_{n_c} based on ξ_{n_c-1}.

This method cannot provide an analytical expression for the transition from ξ_{i-1} to ξ_i since it involves unpredictable times of sample generation. However the conceptual rule of MCMC is that each accepted sample is a sample of the target distribution, which is achieved by the proper transition probability of x_{k+1} at x given that x_k is at v as

follows:

$$\begin{aligned} p_{x_{k+1}|x_k}(x|v) = \int p^*(x;v)\ min\left(1, \frac{f(x)}{f(v)}\right) dx \cdot p_{x_{k+1}|x_k}(x|v, A) + \\ (1 - \int p^*(x;v)\ min\left(1, \frac{f(x)}{f(v)}\right) dx) \cdot p_{x_{k+1}|x_k}(x|v, \overline{A}) \end{aligned} \tag{3.13}$$

where, A denotes the event that one candidate is accepted ($\overline{A}$ for rejected), $p^*(x;v)$ denotes the proposal distribution and $f(\cdot)$ denotes a PDF. Such that x_k is distributed as the target distribution f_π supposing x_{k-1} is distributed as f_π:

$$\begin{aligned} p_{x_{k+1}}(x) &= \int p_{x_{k+1}|x_k}(x|v) p_{x_k}(v) dv \\ &= \int p_{x_{k+1}|x_k}(x|v) f_\pi(v) dv \\ &= \int p_{x_{k+1}|x_k}(v|x) f_\pi(x) dv \\ &= f_\pi(x) \int p_{x_{k+1}|x_k}(v|x) dv \\ &= f_\pi(x) \end{aligned} \tag{3.14}$$

where $\int p_{x_{k+1}|x_k}(v|x)dv$ is a PDF that equals 1. Because the conditional sampling using the repeated sample generation method [114] cannot provide a proper transition probability, it cannot ensure samples x to describe the target distribution f_π.

3.3.3 Delayed rejection

To improve the acceptance rate, Miao and Ghosn [115] proposed to repeat the sample generation process by a second proposal PDF if the initial candidate is rejected by the first rejection event, thus delaying rejection. The steps of conditional sampling for the i^{th} intermediate event F_i can then be described as follows:

Delayed rejection

1. Propose a random standard Normal move $\alpha \sim N(\mathbf{x}_k, 1)$ in each dimension from each seed $\mathbf{x}_1, ..., \mathbf{x}_k, ..., \mathbf{x}_{n_s}$, or alternatively in an adaptive manner: a Normal move using the variance of the current sampling seeds $\mathbf{x}_1, ..., \mathbf{x}_{n_s}$;

2. Accept the candidate movement $\xi_1 = \alpha$ with probability $min\{1, \frac{\phi(\alpha)}{\phi(\mathbf{x}_k)}\}$, and go straight to step 5, otherwise go to step 3;

3. Propose another move e.g. $\alpha' \sim U(\mathbf{x}_k, 1)$ in each dimension from each seed $\mathbf{x}_1, ..., \mathbf{x}_k, ..., \mathbf{x}_{n_s}$;

4. Accept the candidate movement $\xi_1 = \alpha'$ with probability $min\{1, \frac{\phi(\alpha')}{\phi(\mathbf{x}_k)}\}$, otherwise stay at $\mathbf{x}_k$;

5. Remain at $\mathbf{x}_k$ if if the movement does not belong to the intermediate level F_i $\xi_1 \notin F_i$;

6. Sequentially conduct the conditional sampling: obtaining ξ_3 based on ξ_2, obtaining ξ_4 based on ξ_3, ..., until obtaining ξ_{n_c} based on ξ_{n_c-1}.

This method allows an analytical expression of the transition PDF. However one can see intuitively that a second chance would not improve much the acceptance rate, which will be illustrated by numerical examples in chapter 4.

3.3.4 Adaptive MCMC with optimal scaling

Roberts et al. [116, 117] found that the optimal efficiency is achieved when the sample acceptance rate is around 0.44. Inspired by this, Papaioannou et al. [113] proposed to scale the standard deviation of the proposal PDF adaptively (using a scaling parameter λ for the standard deviation of the proposal distribution) to stabilize the optimal acceptance probability.

At each level, all n_s seeds are randomly divided into R groups, each with N_a seeds. Given an initial standard deviation for the proposal PDF, the initial N_a seeds are used to generate sample candidates conditioned on them. The value of the scaling parameter λ is determined by the difference between the acceptance rate of the first N_a seeds and the optimal value 0.44. The adapted standard deviation is then plugged into the next group of N_a seeds. The adaptive standard deviation algorithm for conditional sampling in the i^{th} intermediate event is as follows:

Adaptive MCMC with optimal scaling

1. Randomly divide n_s seeds into R groups of N_a samples;
2. For the r^{th} group ($r = 1, ..., R$), set the initial standard deviation of the proposal distribution to $s_i = 1, i = 1, ..., d$, and the initial scaling parameter λ to 0.6 [113];
3. Compute the coefficient: $a_i = \sqrt{1 - (\lambda s_i)^2}$, $i = 1, ..., d$;
4. Generate the sample candidates: $\mathbf{x}'_{k_1} \sim N(a\mathbf{x}_k, \lambda s)$, $k = 1, ..., N_a$, $a = [a_1, ..., a_d]$ and $s = [s_1, ..., s_d]$;
5. Accept $\mathbf{x}'_{k_1}$ if $\mathbf{x}'_{k_1} \in F_i$, otherwise the chain remains at $\mathbf{x}_k$;
6. Sequentially conduct the conditional sampling: obtaining $\mathbf{x}'_{k_3}$ based on $\mathbf{x}'_{k_2}$, obtaining $\mathbf{x}'_{k_4}$ based on $\mathbf{x}'_{k_3}$, , ..., until obtaining $\mathbf{x}'_{k_{n_c}}$ based on $\mathbf{x}'_{k_{n_c-1}}$;
7. Compute the average acceptance rate from $N_a \times n_c$ states of all chains: $A = \frac{\mathcal{I}_F(\cdot)}{N_a \times n_s}$, where $\mathcal{I}_F(\cdot)$ is an indicator function which counts the number of failure samples;
8. Upgrade the adaptive scaling parameter: $\lambda = 10^{(\log 10(\lambda) + \zeta(A - 0.44))}$, with $\zeta = r^{-1/2}$;
9. Repeat steps 2 to 9 until finishing all R groups of n_s seeds.

This method takes advantage of the optimal acceptance rate so that it efficiently improves samples' acceptance rate while maintaining a certain level of 'ergodicity', the ability of generating samples extensively exploring the input space.

3.3.5 Subset-infinity

Au and Patelli [104] extended the conditional sampling to decomposing the sample $\mathbf{x}_i$ of standard Gaussian distribution by an arbitrary number ($1 \leq N \leq \infty$) of i.i.d. standard Gaussian variables. Let

$$\mathbf{x}_i = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} Z_j \tag{3.15}$$

where $Z_j \sim N(0, 1)$. When $N \to \infty$, sample candidates x'_i conditioned on x_i can be generated by a proposal PDF:

$$p^*(x'_i; x_i) = \frac{1}{\sqrt{2\pi} s_i} exp\left[-\frac{1}{2s_i^2}(x'_i - ax_i)^2\right] \tag{3.16}$$

where

$$a_i = 1 - 2\kappa_i \tag{3.17}$$

$$s_i^2 = 4\kappa_i - 4\kappa_i^2 \tag{3.18}$$

$$\kappa_i = \int_0^\infty w^2 \Phi(-\frac{w}{2}) p_i^*(w) dw \tag{3.19}$$

It can be shown that $a_i^2 + s_i^2 = 1$, and $0 \leq \kappa \leq 1$, hence $a_i \in [-1, 1]$ and $s_i \in [0, 1]$. To sample in the i^{th} intermediate event F_i, the conditional sampling algorithm for Subset-infinity is:

Subset-infinity

1. Assign values for parameters of the proposal PDF: e.g. $s_i = 0.5$ (suggested by [118]), $a_i = \sqrt{1 - s_i^2}$, $i = 1, ..., d$;
2. Generate the sample candidate dimension-wise: $\mathbf{x}'_{k_1} \sim N(a_i \mathbf{x}_k, s_i)$, $k = 1, ..., n_s$;
3. Accept $\mathbf{x}'_{k_1}$ if $\mathbf{x}'_{k_1} \in F_i$, otherwise the chain remains at $\mathbf{x}_k$;
4. Sequentially conduct the conditional sampling: obtaining $\mathbf{x}'_{k_3}$ based on $\mathbf{x}'_{k_2}$, obtaining $\mathbf{x}'_{k_4}$ based on $\mathbf{x}'_{k_3}$, ..., until obtaining $\mathbf{x}'_{k_{n_c}}$ based on $\mathbf{x}'_{k_{n_c-1}}$.

A version of this method was also proposed by Papaioannou et. al. [113]. They consider samples on a chain as multivariate Normal distributions, with a diagonal covariance matrix, of which the i^{th} term equal to s_i. In their work, this method is called 'conditional sampling'. To distinguish from the whole class of MCMC conditional sampling schemes, in this dissertation this method is referred to as 'Subset-infinity'. Papaioannou et. al. also presented that Subset-infinity and adaptive MCMC with optimal scaling have similar accuracy and efficiency—both method bypass the first rejection event, which greatly improves the acceptance rate. Yet adaptive MCMC with optimal scaling is more efficient with adaptive scaling that guarantees the ergodicity property.

3.4 Subset Simulation example

The following example was presented in [104]. Consider a moment resisting frame (Figure 3.4) with 7 input variables (d=7). The units for all variables are kN. Let $\theta_1 \sim N(80, 8^2), ..., \theta_5 \sim N(80, 8^2)$ be the moment capacities at joints and let $\theta_6 \sim N(20, 6^2), \theta_7 \sim N(25, 7.5^2)$ be the loads. The performance function takes the maximum

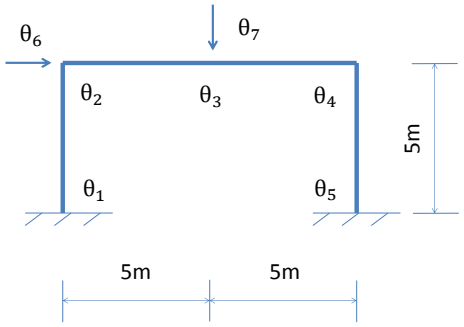


FIGURE 3.4: A moment resisting frame with independent Normal variables $\theta_1, ..., \theta_7$..

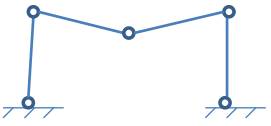
of its three collapse conditions (Figure 3.5): $y = max(g_1, g_2, g_3)$. The critical value y^*

Failure Mode 1

$$g_1 = 5\theta_6 + 5\theta_7 - (\theta_1 + 2\theta_2 + 2\theta_4 + \theta_5)$$

Failure Mode 2

$$g_2 = 5\theta_6 - (\theta_1 + 2\theta_2 + \theta_4 + \theta_5)$$



Failure Mode 3

$$g_3 = 5\theta_7 - (\theta_2 + 2\theta_3 + \theta_4)$$

FIGURE 3.5: Three failure modes of this moment resisting frame.

for the response variable y takes $y^* = 0$, which means that when the loads are larger than the moment capacity the frame fails.

The parameters for Subset Simulation are the number of samples $n_L = 1000$, the level probability $p_L = 0.1$, the number of seeds $n_s = n_L \times p_L = 100$ and the number of conditional samples generated by MCMC for each chain $n_c = (1 - p_L)/p_L = 9$. The proposal PDF is the standard Normal distribution. 100 independent runs of Subset Simulation yields $\hat{p_F} = 7.4^{-7}$, with levels $L = 6$. The failure samples and the conditional sampling process of Subset Simulation for loads θ_6, θ_7 are illustrated in Figure 3.6.

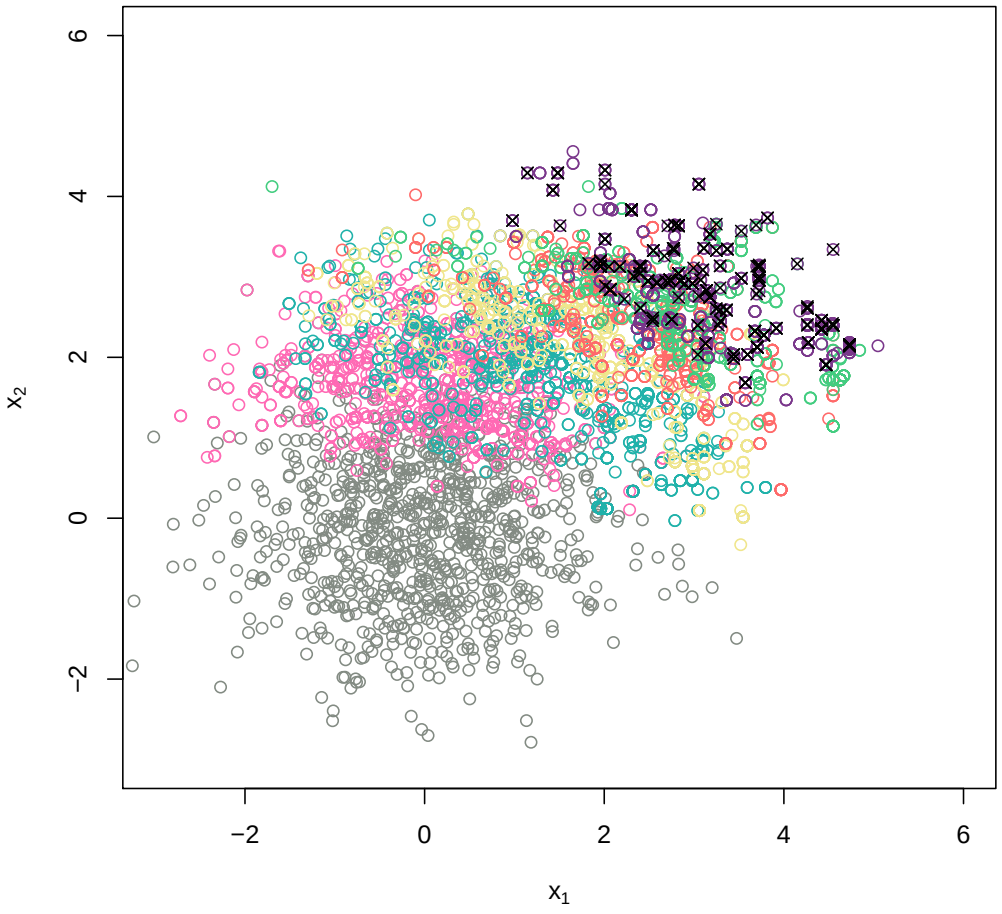


FIGURE 3.6: Subset Simulation samples in different levels. The grey circles are the samples of the original standard Normal distribution. Colored circles are the samples of the intermediate conditional distributions. The failure samples are denoted by '×' symbols.

Consider the following variation from the example in [104], the PDFs of the input variables are changed to Normal so that for comparison purposes, the probability of failure p_F can be calculated analytically by taking variables $z_1 = 5\theta_1 + 5\theta_7 - (\theta 1 + 2\theta_3 + 2\theta_4 + \theta_5)$, $z_2 = 5\theta_6 - (\theta_1 + 2\theta_2 + \theta_4 + \theta_5)$, $z_3 = 5\theta_7 - (2\theta_2 + \theta_3 + \theta_4)$, such that $z_1 \sim N(-255, 54.3^2)$,

$z_2 \sim N(-300, 36.7^2)$, $z_3 \sim N(-275, 42.3^2)$, hence $\frac{z_1+255}{54.3} \sim N(0,1)$, $\frac{z_2+300}{36.7} \sim N(0,1)$, $\frac{z_3+275}{42.3} \sim N(0,1)$, which can be calculated by the CDF of the standard Gaussian distribution:

$$P(z_i > y^*) = P(\frac{z_i}{\sigma_i} > \frac{-\mu_i}{\sigma_i}) = 1 - \Phi(\frac{-\mu_i}{\sigma_i}),\ i = 1, 2, 3 \tag{3.20}$$

where, Φ is the CDF of the standard Gaussian distribution, μ_i is the mean value of z_i, and σ_i is the standard deviation of z_i. The analytical solution of the probability of each failure mode is $p_{F_{g_1}} = 1.3 \times 10^{-6}$, $p_{F_{g_2}} = 1.5 \times 10^{-16}$, $p_{F_{g_3}} = 4 \times 10^{-11}$. Thus, the failure probability of $P(F\{y = max(g_1, g_2, g_3) > y^*\}) = 1.3 \times 10^{-6}$. The estimated probability of failure from Subset Simulation 0.74×10^{-6} is very close. Subset Simulation used $N = n_L + L(n_L - n_s) = 6400$ number of samples in total and in the end it possesses about 700 distinct failure samples. However, Direct Monte Carlo would need about 10^6 samples to get only one failure sample, and therefore not feasible to correctly estimate the probability of failure for this example.

3.5 Summary

In this chapter, Subset Simulation was reviewed. It was originally developed to compute rare failure probabilities in engineering reliability analysis. The conditional sampling technique makes sampling for rare events feasible, which is illustrated by a frame resistance example. Different MCMC sampling schemes are discussed to more efficiently populate the target space. Based on sampling schemes in this chapter, Subset Simulation is proposed to solve the sampling problem for HM in the next chapter, since multiple common features can be found in HM and reliability analysis. A natural analogy can be established between the non-implausible domain for HM and a failure event for reliability analysis, since both of them require the definition of a 'threshold'—reliability analysis uses safety laws to define a failure event and HM uses an implausibility measure to define a non-implausible domain. In addition, a non-implausible domain can diminish to orders of magnitude smaller than the original input space, which is difficult to sample just as a rare failure event. Last but not least, Subset Simulation can be used to 'black box' performance functions and HM can be used to 'black box' simulators, where there

is not much information about the characteristics of the system behavior. Hence, using Subset Simulation to sample the non-implausible domain for HM do not limit the type of application.

Chapter 4

Subset Simulation for Non-implausible Sampling

4.1 Introduction

HM iteratively cuts down the input space to find the fitting input domain that provides a reasonable match between model output and experimental data. It sequentially refocuses on the non-implausible domain that has a potential to reproduce the observation, which can significantly reduce the computational time of running complex simulators on a large number of samples covering a high-dimensional input space. Pointed out by Vernon et al. [45], this continued refocusing is powerful, but the only way to determine whether each sample lies in the non-implausible domain is to apply the implausibility measure in turn. Therefore, each wave requires a sampling step that can fastly screen the input space to generate non-implausible samples. It is challenging when the non-implausible domain is with nontrivial topology, disconnected, or orders of magnitude smaller than the original input space.

As it has been mentioned, there exist natural analogies between a non-implausible domain in HM and a failure set in reliability analysis. Firstly, both of them require a sampling 'threshold'. Second, both of them can treat the system of interest as a 'black box'. Finally, for both the target space can be orders of magnitude smaller than the

original space. Hence, Subset Simulation is proposed to generate non-implausible samples. It progressively decomposes a rare event, here is the non-implausible set for HM, which has a small probability of occurrence, into sequential less rare intermediate events. Samples providing better non-implausibilities in intermediate events remain as seeds to generate more samples towards the target space. To adapt Subset Simulation for non-implausible sampling, some modifications to the method are needed. The sampling criterion of the critical response value $y > y^*$ should be substituted by the implausibility threshold $I(x) < 3$ and the performance function should be substituted by the implausibility measure.

This chapter introduces the implementation of Subset Simulation for non-implausible sampling in HM (SSHM). Low dimensional examples visually show the potential of this proposed method, with non-implausible domains having disconnected nontrivial topologies and being orders of magnitude smaller than the original input space. Then the complete implementation of the proposed SSHM is illustrated by a 1-D example and a 10-D example respectively. The performance of different MCMC sampling schemes for Subset Simulation is compared in the context of SSHM.

This chapter also considers further improvements of the proposed SSHM framework. As has been shown, a use of iterative succession of emulators requires the refinement of the emulator at each wave, so that the variance of the estimated model output can be gradually reduced and the non-implausible domain can be further cut down. This is done through adding new training data on the non-implausible domain. Screening the input space to select new training data located on the non-implausible domain can be challenging since the non-implausible domain can be orders of magnitude smaller than the original input space. Considering that SSHM always refocuses on the non-implausible domain, a strategy for selecting new training data is proposed. Non-implausible samples defined by the current wave can be reused as new training data in the next wave. However the non-implausible domain may be disconnected and the refinement is expected to happen on each region. Hence, the use of a classification method is proposed to detect each region of the non-implausible domain and select samples from them respectively.

The implementation of using a classification method to help to generate new training data is illustrated by a numerical example.

4.2 The implementation steps of SSHM

Subset Simulation is a reliability-based sampling strategy. It was designed to find samples with failure probability. In order to adapt Subset Simulation to find samples with non-implausibility in HM, some modifications to the method are required. Instead of evaluating output response, SSHM evaluating implausibility. The decomposition of intermediate levels for Subset Simulation applied in HM is as follows. In each level, Subset Simulation needs to select n_s seeds with the smallest implausibilities. The intermediate threshold is now defined as:

$$Y_{SS_i} = \frac{I_i(x_{n_s}) + I_i(x_{n_s})}{2} \tag{4.1}$$

where $I_i(x_{n_s})$ is the n_s smallest implausibility measure of the seeds for level i. MCMC is used to generate conditional samples that belong to the corresponding level. The stopping condition for the decomposition in Subset Simulation now is to have enough non-implausible samples in a level:

$$n_F = \frac{\sum_{k=1}^{n_L} \mathcal{I}_{I_{\mathbf{x}\in(I<3)}}}{n_L} > p_L \tag{4.2}$$

where n_F is the proportion of non-implausible samples out of all samples in a level and $\mathcal{I}_I(\cdot)$ is an indicator function that counts the number of samples in the relevant set.

The implementation steps of SSHM are constructed by two nested loops. At the beginning, a Latin hypercube sampling is needed to provide approximately evenly scattered sample points in the design space, so that adequate information can be obtained throughout limited samples in model input space. Then the outer loop—the refocusing process of HM begins. The simulator under calibration is run at each sample point, producing training data for an emulator. The emulator is constructed to make inference about the simulator's output at more other unevaluated input points to reduce computational

expenses. An implausibility measure that takes into account diverse sources of uncertainties are applied to rule out implausible samples. The refocusing ends at that enough non-implausible samples are obtained or the uncertainty provided by the emulator is smaller than the rest uncertainties, where the non-implausible domain is unlikely to decrease in the next wave. If the stopping criteria are not met, a new wave starts with applying the modified Subset Simulation to sample from the reduced non-implausible domain. Then the inner loop—the decomposition of the non-implausible domain begins. At each level of Subset Simulation, the first n_L samples with the smallest implausibility measure are selected as seeds. The intermediate threshold for defining this level is obtained by Equation 4.1. New samples are generated using MCMC towards the targeted non-implausible space. The proportion of the non-implausible samples in this level is calculated. The decomposition ends at that Equation 4.2 is satisfied. After applying Subset Simulation, this wave of HM is completed by running the simulator to obtain more training data, refining the emulator and applying the implausibility test to rule out more implausible domain. Figure 4.1 summarizes the diagram of SSHM.

4.3 Benchmark criteria for sampling schemes

As aforementioned, Subset Simulation includes variants of MCMC sampling schemes. To compare their performance when sampling in the framework of SSHM, benchmark criteria suitable for calibration purposes should be considered:

1. Distinct non-implausible samples. HM is a model calibration method, which aims to find fitting input samples that can provide a match between model outputs and observations. However, MCMC sampling schemes involve rejection events of sample candidates. If sample candidates have larger implausibility they may get rejected, or if sample candidates do not belong to the corresponding intermediate level they are rejected. In those cases, the samples do not move towards the target space but remain at the same coordinates, which is a disadvantage for detecting the non-implausible domain. The capability of generating more distinct

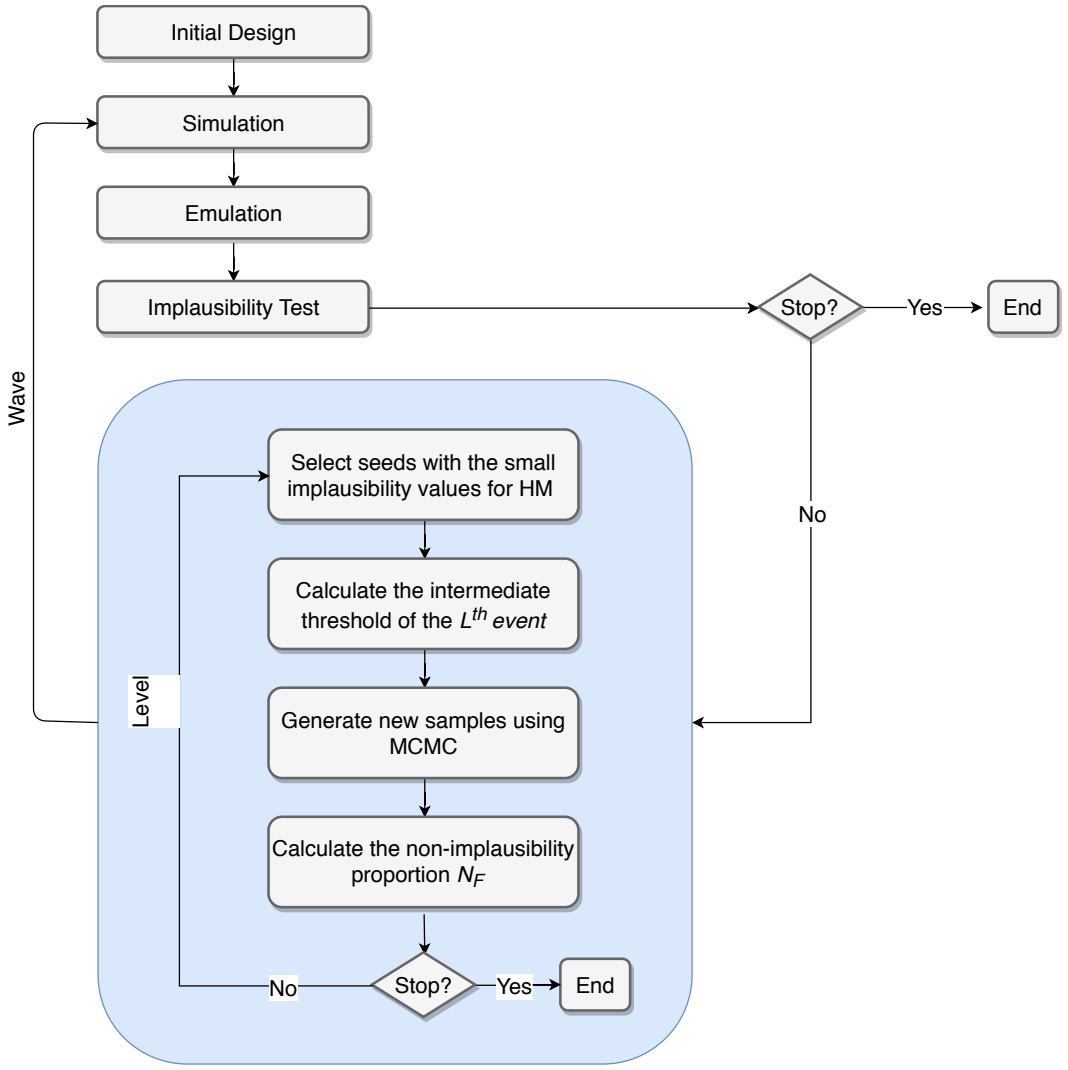


FIGURE 4.1: The diagram of SSHM. Subset Simulation is adapted to sample from the reduced non-implausible input domain. Diverse MCMC sampling schemes were discussed in chapter 3.

non-implausible samples helps HM to more efficiently identify a non-implausible domain.

2. HM waves. The number of HM waves represents the computation time. Less waves under the same stopping criterion for HM represent a fast elimination of the implausible domain, and vise versa.

3. Non-implausible probability. Analogously to that Subset Simulation should give a correct estimate for the probability of failure in reliability analysis, Subset Simulation should also give a correct estimate for the probability of non-implausible matches. Similarly as the failure probability (Equation 3.11), the non-implausible

probability is estimated as:

$$\hat{P}(y > y^*) = p_L^k \frac{\sum_{k=1}^{n_L} \mathcal{I}_{I_{\mathbf{x}\in(I<3)}}}{n_L} \tag{4.3}$$

It helps to know whether the observed data are frequent values for the physical quantity and hence helps to understand the physical process under study.

4. Coefficient of variation (c.o.v.). It quantifies the correlation among samples in the same level and across different levels for Subset Simulation. A low value of c.o.v. means less rejection of sample candidates, more efficient exploration of the input space, and hence a more accurate estimator without increasing the sample size [114]. There does not exist a formula to describe correlation across different levels, but c.o.v. can be bounded by the fully correlated case using Equation 4.4 and the independent case using Equation 4.5 [87].

$$\alpha_{L_i} = (\sum_{j=1}^{i} \delta_j^2)^{1/2} \tag{4.4}$$

$$\alpha_{U_i} = (\sum_{j=1}^{i} \sum_{k=1}^{i} \delta_j \delta_k)^{1/2} \tag{4.5}$$

where,

$$\delta_j^2 = \frac{1 - p_L}{p_L N}(1 + \gamma_j) \tag{4.6}$$

$$\begin{aligned} \gamma_1 &= 0 \\ \gamma_i &= 2 \sum_{k=1}^{n_c-1} (1 - \frac{k}{n_c}) \rho_i(k) \quad i = 2, ..., m \end{aligned} \tag{4.7}$$

$$\rho_i(k) \approx \frac{1}{p_L(1-p_L)} \left[\frac{1}{n_s(n_c-k)} \left[\sum_{j=1}^{n_s} \sum_{r=1}^{n_c-k} I(Y_{j,r}^{(i-1)} > Y_{SS_L}) I(Y_{j,r+k}^{(i-1)} > Y_{SS_L}) \right] - p_L^2 \right] \tag{4.8}$$

where, γ_i is called the sample correlation factor for level $i-1$ and $\rho_L(k)$ is the correlation coefficient of samples k-steps apart along a chain. Therefore, the c.o.v. of the non-implausible probability for level i is obtained by:

$$\alpha_{L_i} < \alpha_i < \alpha_{U_i} \tag{4.9}$$

4.4 Examples for sampling w.r.t. implausibility functions

The function of the implausibility measure $I(x)$ may appear a complex form, resulting in several modes, which leads to a nontrivial topology and disconnected regions for a non-implausible domain [47]. This is shown by a two-dimensional numerical example. A three-dimensional example then shows that the non-implausible domain can be only a tiny fraction of the design space. Both of them provide visual illustrations for that Subset Simulation manages to sample on the corresponding non-implausible domain. The performance of different MCMC sampling schemes is compared according to the previously discussed benchmark criteria within the HM framework.

4.4.1 A 2-D implausibility function

Let the initial input space be the plane $[-10, 10]^2$. The implausibility function [119] is as follows:

$$I(\mathbf{x}) = \frac{500x_1x_2}{x_1^2 + x_2^2} + 250 \tag{4.10}$$

The surface for this model is shown in Figure 4.2. The mission for Subset Simulation is to

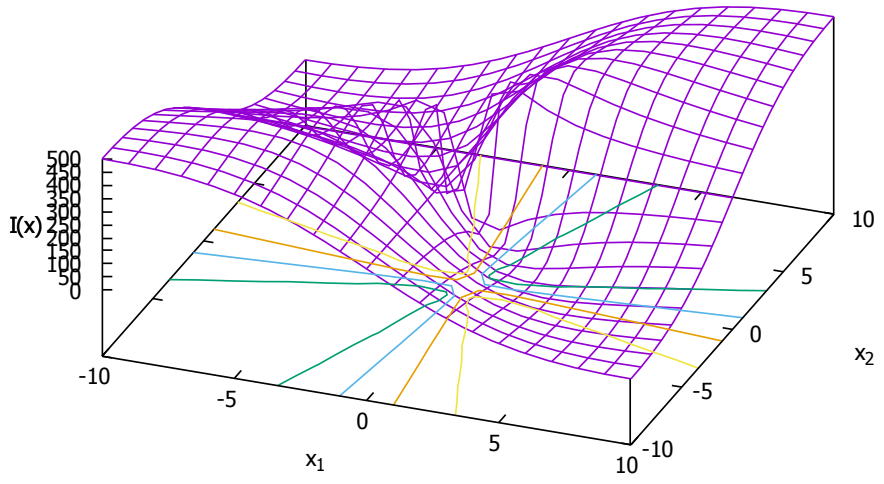


FIGURE 4.2: Surface of the implausibility function with different implausibility contours in intervals of 100: respectively $I(\mathbf{x}) = 100$ (the yellow contour), 200 (the red contour), 300 (the blue contour), and 400 (the green contour).

populate samples on the non-implausible domain, defined by $I(\mathbf{x}) \leq 3$. The parameters are chosen according to the four rules introduced in section 3.1. The number of samples in each level strikes a balance between the computation speed and the coverage of the input space $n_L = 5000$ and the level probability strikes a balance between the required number of levels and efficient conditional sampling $p_L = 0.1$.

From a single run of Subset Simulation, Figure 4.3 shows the samples in the last level.

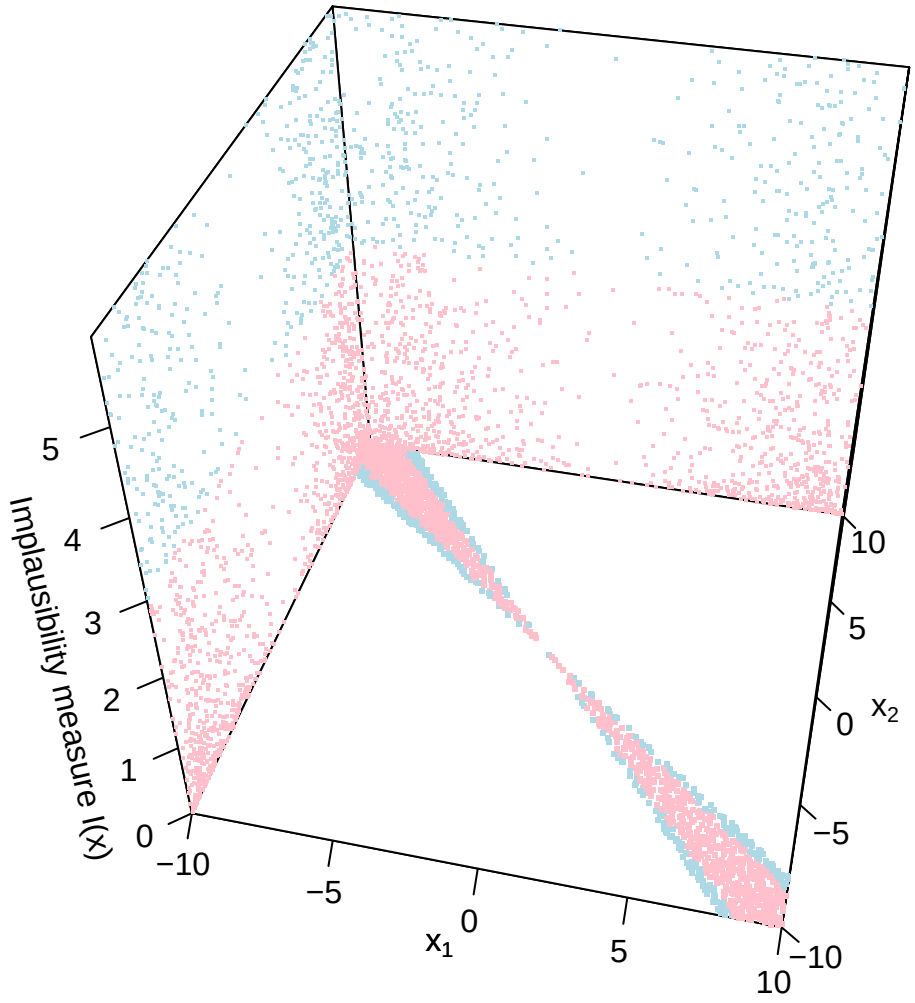


FIGURE 4.3: Subset Simulation samples in the last level. Pink dots correspond to non-implausible samples ($I(\mathbf{x}) \leq 3$) and blue dots implausible samples ($I(\mathbf{x}) > 3$).

Table 4.1 compares the performance of different MCMC algorithms. Data in Table 4.1 show the mean values through independent 100 runs of Subset Simulation for each

MCMC algorithm.

MCMC algorithms	Distinct non-implausible samples	Levels [a]	Non-implausible probability	c.o.v.
MMA	1583	1	0.071	0.0118
MMA (adaptive)	1466	1	0.073	0.0124
Repeated sample generation	1422	1	0.075	0.0123
Repeated sample generation (adaptive)	1304	1	0.075	0.0129
Delayed rejection	1475	1	0.073	0.0123
Delayed rejection (adaptive)	1420	1	0.073	0.0124
Subset-infinity (s=0.5)	2015	1	0.067	0.0103
Adaptive optimal scaling	1041	1	0.070	0.0139

[a] This example does not involve the wave of HM yet, thus the number of levels for Subset Simulation represents the benchmark criterion for computation time.

TABLE 4.1: The performance difference among different MCMC sampling schemes, with parameters for Subset Simulation: $p_L = 0.1$ and $n_L = 5000$.

In this low-dimensional example with the non-implausible domain not relatively small (about 7%-7.5%) than the original input space, the performance of different MCMC sampling schemes is similar.

4.4.2 A 3-D implausibility function

Let the initial input space be a cube $[-20, 40]^3$. The following implausibility function comes from Williamson and Vernon [47]:

$$I(\mathbf{x}) = \frac{1}{10}(\sqrt{u^\intercal \sigma^{-1} u} + \frac{x_3^2}{0.04^2}) \tag{4.11}$$

where $u = \begin{pmatrix} (x_1-2)^2 - 3 \\ (x_2-2)^2 - 3 \end{pmatrix}$, $\sigma = \frac{1}{2^{12}} \begin{pmatrix} 1 & -0.97 \\ -0.97 & 1 \end{pmatrix}$.

The non-implausible domain is defined by $I(\mathbf{x}) \leq 3$, resulting in the dark blue dots shown by Figure 4.4. The implausible domain corresponds to the rest of the cubic input space. Subset Simulation decomposes the rare non-implausible set into sequential less rare intermediate sets, based on which the conditional sampling was conducted. For example, green and yellow dots denote samples from two different intermediate levels for Subset Simulation.

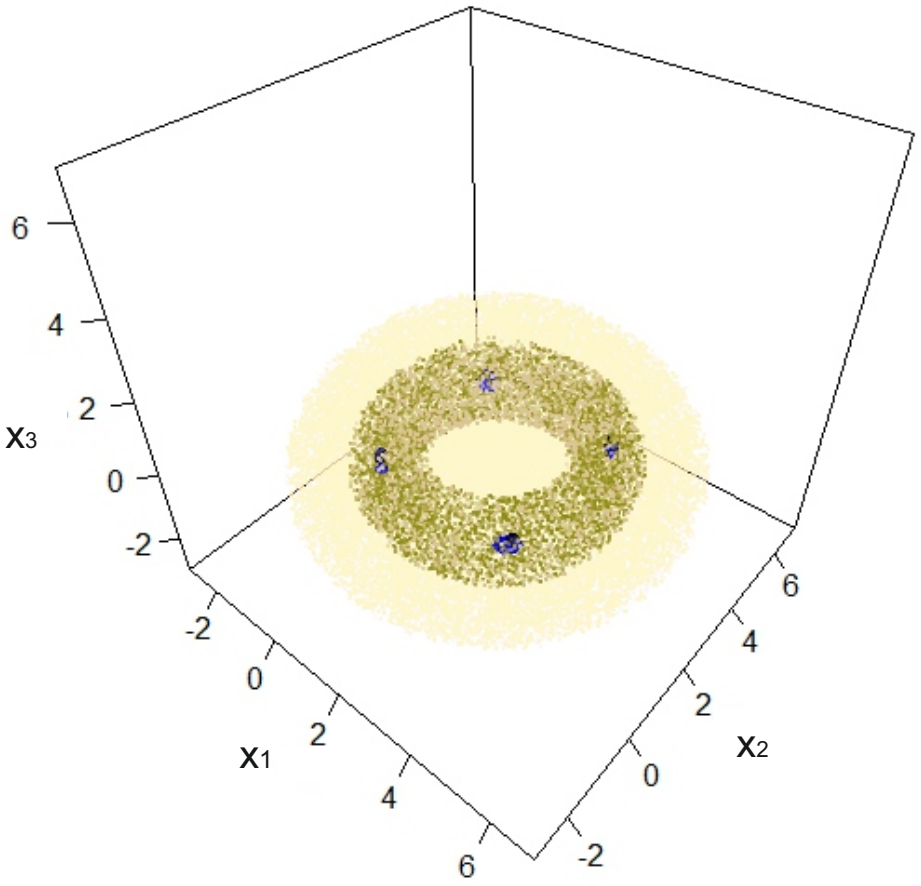


FIGURE 4.4: Subset Simulation samples. Dark blue dots denote the non-implausible samples. Green and yellow dots denote samples from different intermediate levels for Subset Simulation.

Table 4.2 compares the performance of different MCMC algorithms. Data in the table show the mean values through independent 100 runs of Subset Simulation for each

MCMC algorithm.

MCMC algorithms	Distinct non-implausible samples	Levels [a]	Non-implausible probability	c.o.v.
MMA	90	7	1.05×10^{-7}	0.0039-0.0099
MMA (adaptive)	200	7	6.27×10^{-8}	0.0033-0.0080
Repeated sample generation (adaptive)	120	7	5.33×10^{-8}	0.0034-0.0082
Delayed rejection	17	10	2.20×10^{-9}	0.0038-0.0100
Delayed rejection (adaptive)	151	7	6.33×10^{-8}	0.0034-0.0082
Subset-infinity (s=0.1)	23	7	5.62×10^{-8}	0.0037-0.0093
Subset-infinity (s=0.01)	2004	7	5.57×10^{-8}	0.0036-0.0092
Adaptive optimal scaling	601	7	5.46×10^{-8}	0.0035-0.0088

[a] This example does not involve the wave of HM yet, thus the number of levels for Subset Simulation represents the benchmark criterion for computation time.

TABLE 4.2: The performance difference among different MCMC sampling schemes, with parameters for Subset Simulation: $p_L = 0.1$ and $n_L = 50000$.

In this case, Subset-infinity shows the flexibility and efficiency of simply changing the standard deviation parameter s to get more distinct non-implausible samples depending on the problem. On the other hand, although the repeated sample generation makes all movement pass the first rejection event of MCMC, a majority number of them are rejected by the second rejection event. Therefore the repeated sample generation does not objectively improve the sample acceptance probability as it intuitively suggests. It fails to find the target space even with 20 levels, far less efficient than other sampling schemes. Last but not least, presented by [47], the non-implausible probability is less than 6×10^{-8}. It can be seen that all above MCMC sampling schemes provide a close estimate.

4.5 Examples to illustrate the implementation of SSHM

In this section, a simple one dimensional HM example is illustrated to explain every step of the proposed SSHM. Then a 10-D example focuses on the comparison of different sampling schemes in the context of SSHM.

4.5.1 A 1-D model

A physical process y is assumed being modeled by a function: $g(x) = \sin(4x)+x^2$. Let the calibrating input domain be in one dimension on the interval [-1.0, 0.5]. SSHM is used to calibrate this deterministic model. For illustration purposes, let the model discrepancy arbitrarily take a value of 0.005. The observation is assumed to be -0.5, with a standard deviation of 0.06. Uniformly distributed sample points $x = [-1, -0.7, -0.4, -0.1, 0.2, 0.5]$ are the initial design of experiment to run the simulation model. The corresponding simulation output is $g(\mathbf{x})$: [1.7568, 0.1550, -0.8396, -0.3794, 0.7574, 1.1593].

The Gaussian process emulator interpolates 1000 samples to explore the input domain. Assuming there is not much information of the model beforehand, the prior of the emulation is set to be a constant: $h(\mathbf{x}) = [1]$, which does not suggest any trends of the output's response. The $Matérn(5/2)$ correlation function [72] $c(\mathbf{x}, \mathbf{x}') = \left(1 + \sqrt{5}\frac{|\mathbf{x}-\mathbf{x}'|}{\theta} + \frac{5}{3}(\frac{\mathbf{x}-\mathbf{x}'}{\theta})^2\right) e^{\frac{-\sqrt{5}|\mathbf{x}-\mathbf{x}'|}{\theta}}$ is applied due to its moderate smoothing degree.

Figure 4.5 shows the result of the first wave of HM. The six training samples are presented by the red dots; the curve of the emulator output is consistent with the curve of the simulator output; the red dash curves are the 95% credible intervals for the emulation; the three parallels are the specified observation plus or minus its measurement error.

From the figure, it can be seen that the emulator's output fits well with the simulator' output. In a realistic calibration case, the emulator needs to be validated. The cross validation method is applied, which uses a 'leave-one-out' strategy for the six training samples in turn. It is shown by Figure 4.6. The first panel is a direct comparison for the values of the simulation output and the emulation output. The second panel shows the standardized residuals $d_{s_i} = d_i/\sqrt{\sigma^2}$, where d_i is obtained by omitting the ith

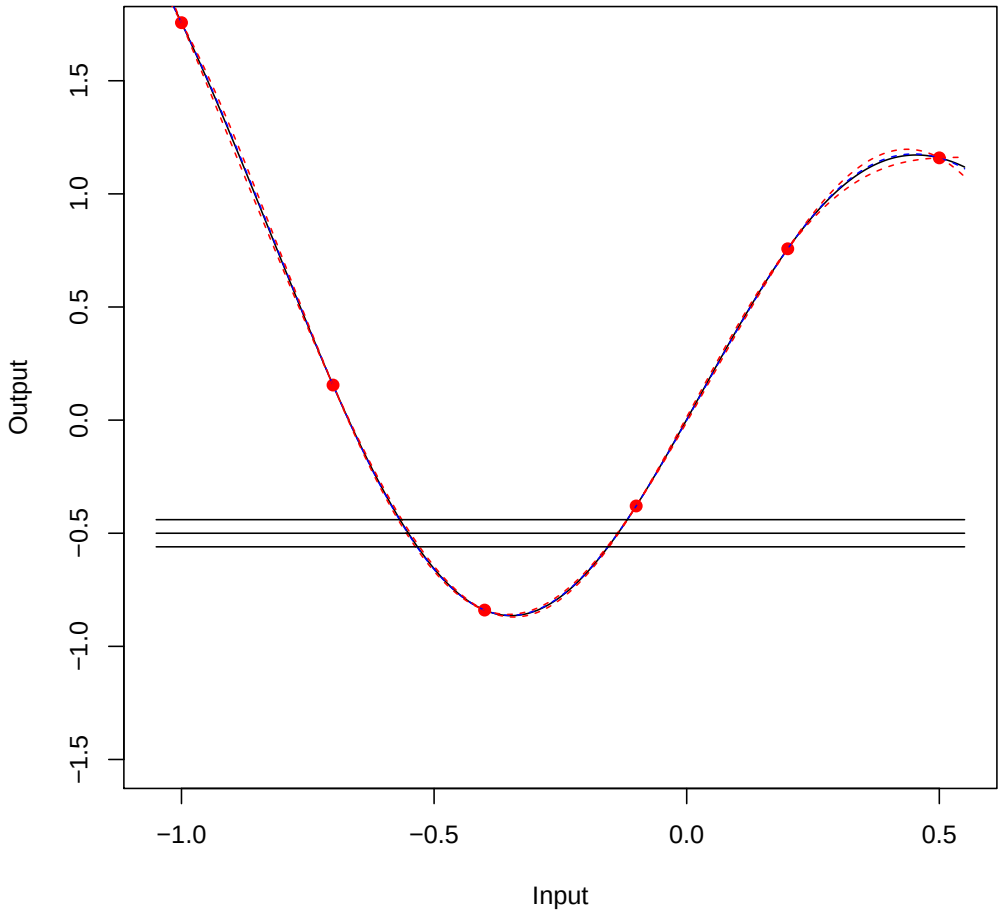


FIGURE 4.5: Based on the simulation data, the emulation's output with confidence intervals for the first wave. The observation with measurement uncertainty is -0.5 ± 0.06.

training sample x_i and calculating the difference between the simulation output $g(\mathbf{x}_{\sim i})$ and the emulation output $E^*(g(\mathbf{x}_{\sim i}))$ at that sample $d_i = E^*(g(\mathbf{x}_{\sim i})) - g(\mathbf{x}_{\sim i})$, and σ^2 is the variance of the residuals d_i. The standardized residual of an appropriate emulator should follow a standard Normal distribution, therefore the values should be within ± 3 according to Pukelsheim three sigma rule [81]. The third panel is the quantile-quantile plot (QQ plot), which displays quantile differences between the probability distribution of the standardized residual and the standard Normal distribution.

As observed from the validation graphs, the simulation output and emulation output are close, the standardized residuals are small and roughly scattered around zero, and when compared to the standard Normal distribution in the QQ plot, it is of linear consistency.

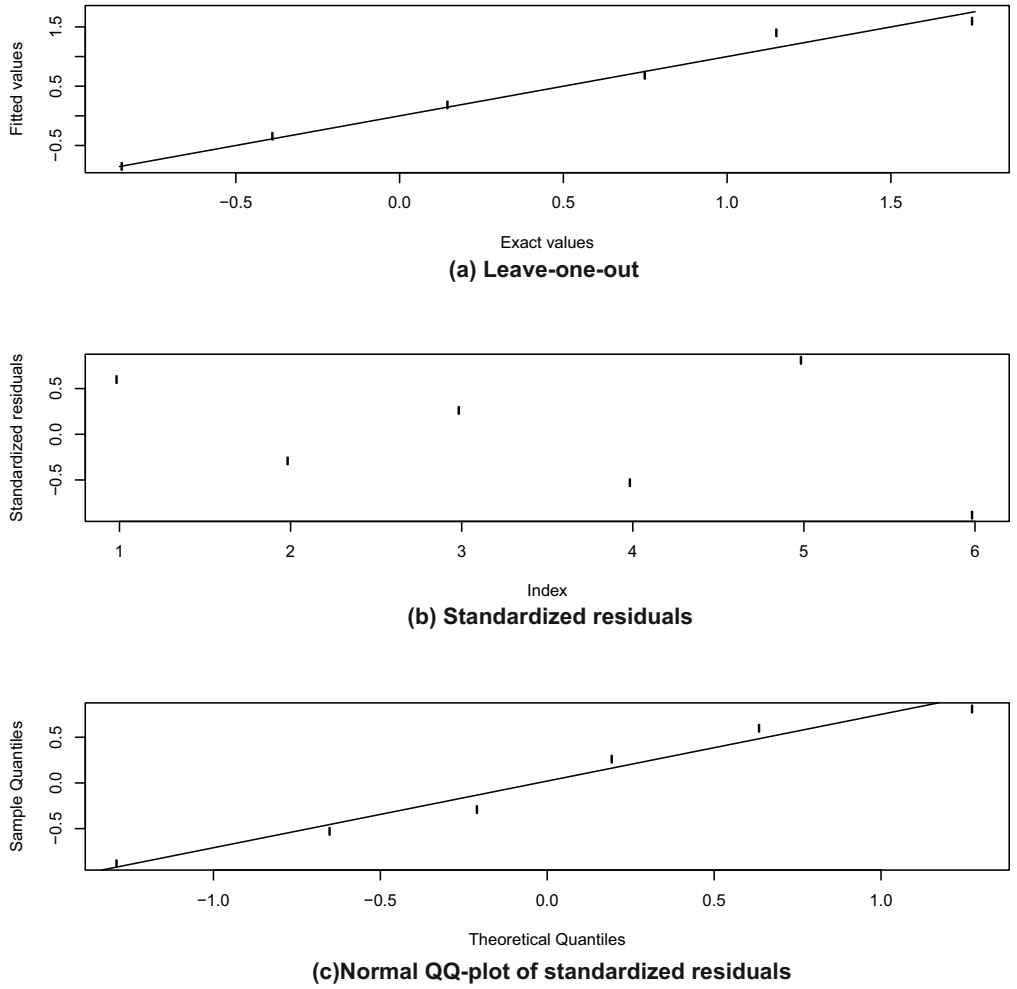


FIGURE 4.6: The cross validation of the emulator for the first wave. An acceptable emulator should have its output in line with the simulator's output, shown by figure (a). The standardized residuals should be small and scattered around zero, shown by figure (b), and follow the standard Normal distribution, shown by figure (c).

Therefore the emulator is acceptable .

After checking the validity of the emulator. The implausibilities of the samples are tested by comparing their output calculated by the emulator with the observation. The uncertainty margins are calculated as follows: $V(\mathbf{x}) = V_o + V_s + V_c + V_m = 0.03^2 + 0 + (\hat{\sigma}\sqrt{c^{**}(\mathbf{x}, \mathbf{x}')})^2 + 0.005$. When the implausible measure $I(\mathbf{x}) = \frac{|-0.5 - E^*(g(\mathbf{x}))|}{V(\mathbf{x})}$ is less than 3, it describes the non-implausible domain after the first wave. As Figure 4.7 shows, the non-implausible intervals are [-0.62, -0.44] and [-0.26, -0.06].

For the illustration purpose to pursue a sequential wave and apply Subset Simulation to sample on the non-implausible domain, the stopping criterion adopted in this example

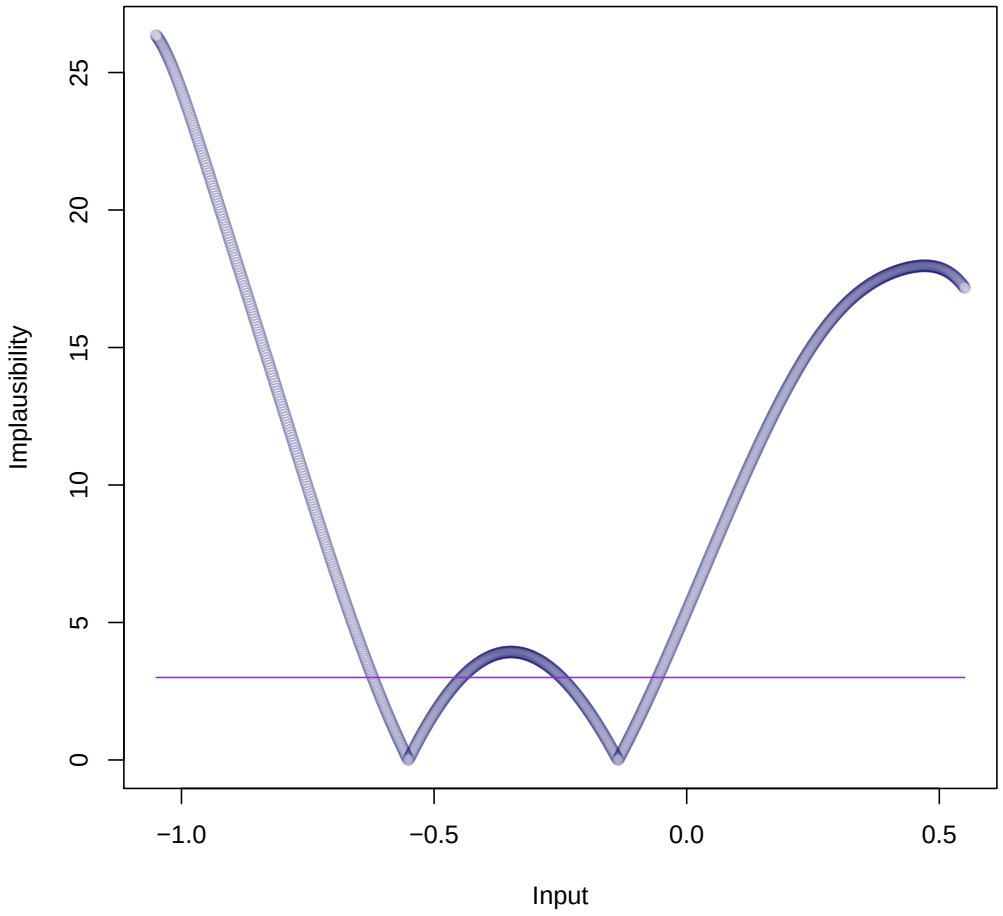


FIGURE 4.7: The implausibility of samples for the first wave.

is to obtain more than 50% non-implausible samples in a wave. The proportion of non-implausible samples out of all samples is 22.9% in the first wave. Hence the sequential wave begins. The emulator can be refined by adding a new training sample on the non-implausible domain. New emulation points are generated by Subset Simulation to refocus on the non-implausible domain. The parameters for the Subset Simulation are chosen according to the four rules introduced in section 3.1. The number of samples in each level strikes a balance between the computation speed and the coverage of the input space $n_L = 900$. The level probability has to be larger than the proportion of the non-implausible samples out of all samples as well as to ensure integer values for the number of seeds $n_s = n_L \times p_L$ and the number of the conditional samples generated by each seed $n_c = (1 - p_L)/p_L$, thus $p_L = 1/3$. In this way, each seed generates two samples in every level, trying to populate towards the target domain.

The emulation for the second wave and the updated training sample are illustrated in Figure 4.8 with the same principles as Figure 4.5. The validation of the emulator is performed via Figure 4.9 and the emulator is accepted.

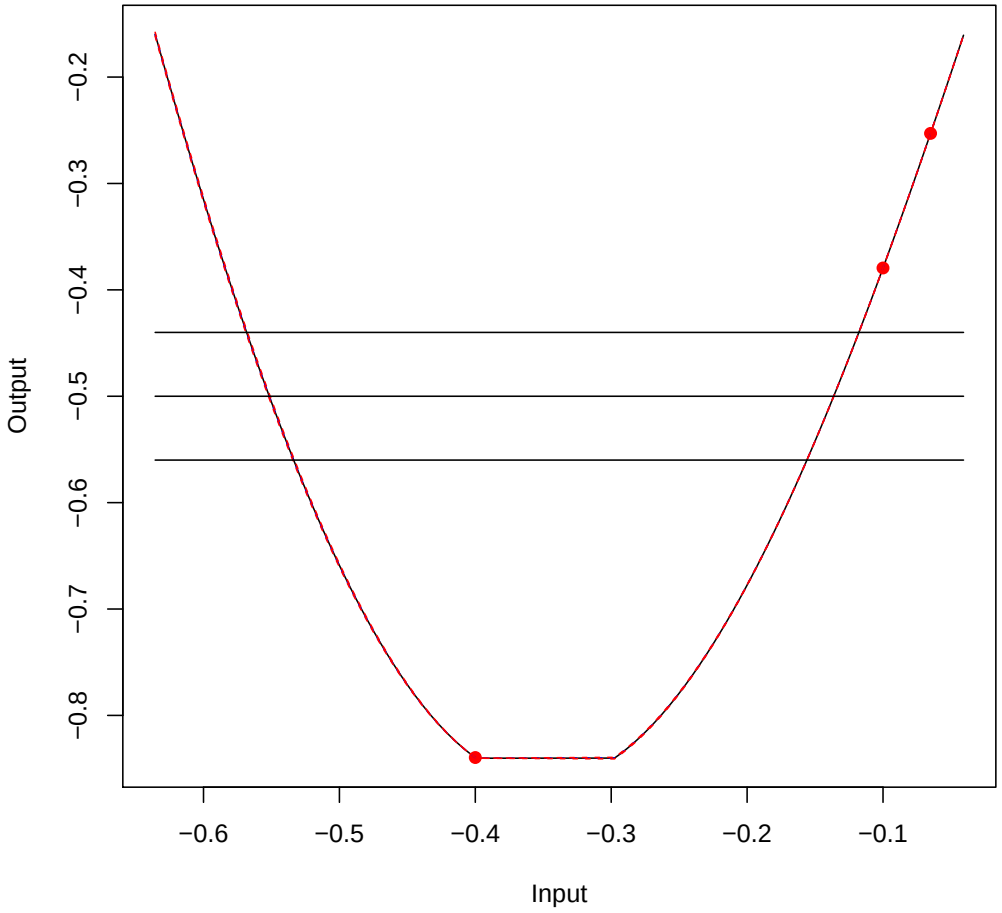


FIGURE 4.8: Based on the simulation data, the emulation's output with confidence intervals for the second wave. The observation with measurement uncertainty is -0.5 ± 0.06.

The non-implausible set consists of the intervals [-0.62, -0.44] and [-0.25, -0.06] for the second wave, as Figure 4.10 shows. The intervals does not narrow much down from the first wave because for such a simple simulation model, the emulator is accurate enough since the first wave. The proportion of the non-implausible sample is 75.2% for the second wave. It satisfies the stopping criterion for HM, hence for this example HM ended after two waves.

The analytical fitting inputs for this simulation function $-0.85 = \sin(4x) + x^2$ on [-1.0,

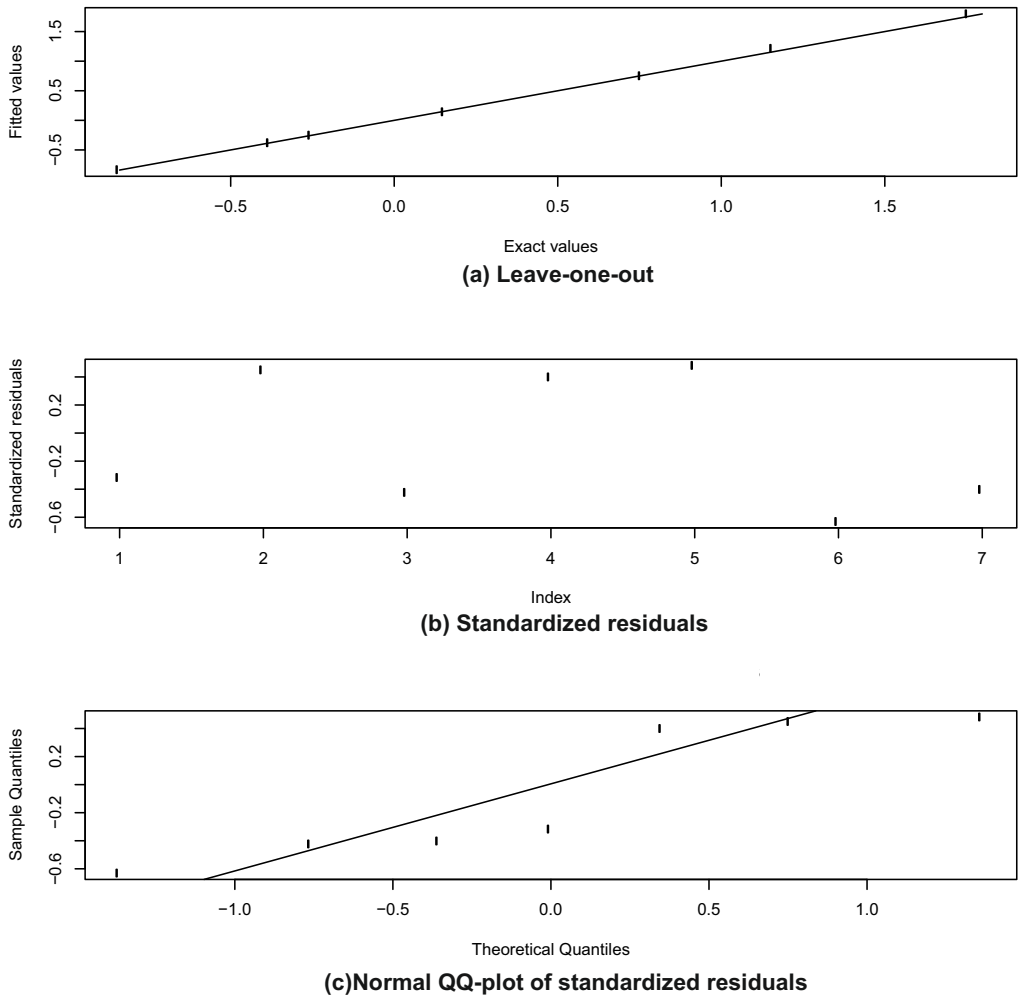


FIGURE 4.9: The cross validation of the emulator for the second wave. An acceptable emulator should have its output in line with the simulator's output, shown by figure (a). The standardized residuals should be small and scattered around zero, shown by figure (b), and follow the standard Normal distribution, shown by figure (c).

0.5] can be calculated as -0.55 and -0.13. It can be seen that they are sufficiently covered by the final non-implausible set consisting of the intervals [-0.62, -0.44] and [-0.25, -0.06] respectively.

4.5.2 A 10-D model

Subset simulation has been proven to be able to generate samples w.r.t. implausibility functions (section 4.4). A 1-D numerical example have visualized the implementation of SSHM, with the result verified by the analytical solution of the problem. Since

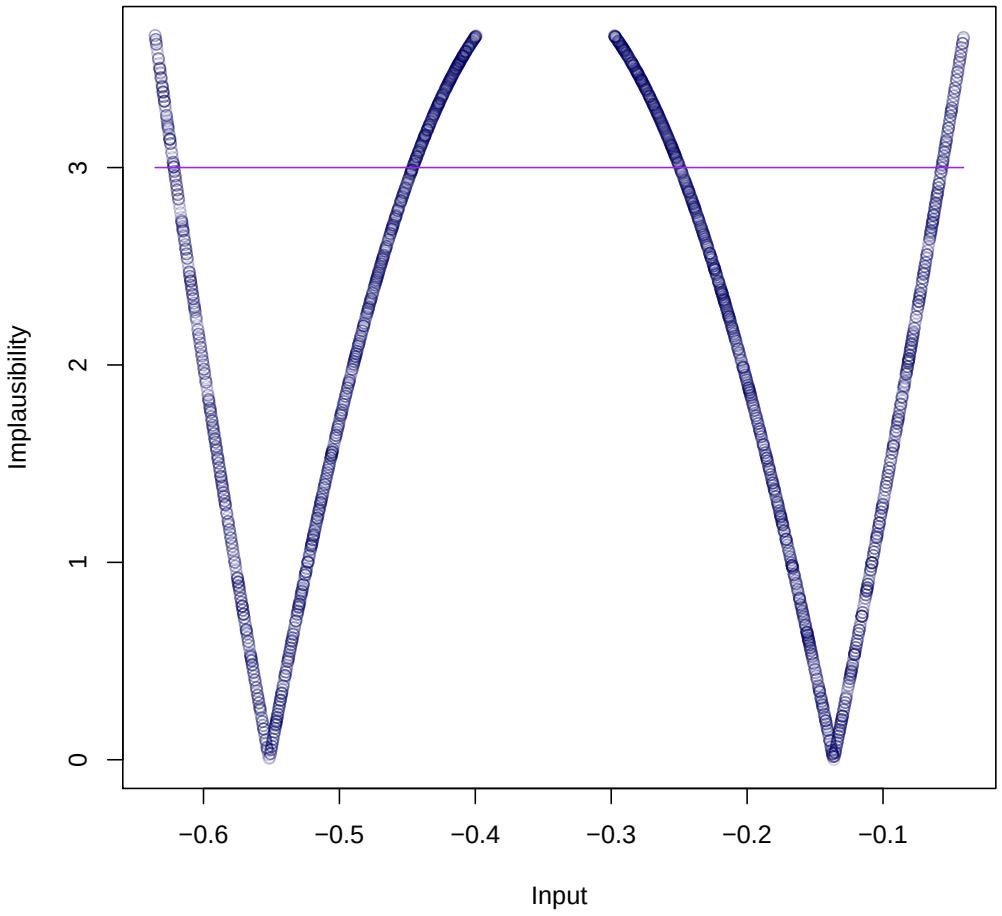


FIGURE 4.10: The implausibility of the samples for the second wave.

the performance difference of MCMC sampling schemes was not remarkable in low-dimensional examples, a 10-D wing weight model for a light aircraft [61] is calibrated:

$$W = 0.036 S_w^{0.758} W_{fw}^{0.0035} \left(\frac{A}{cos^2(\Lambda)}\right)^{0.6} q^{0.006} \lambda^{0.04} \left(\frac{100 t_c}{cos(\Lambda)}\right)^{-0.3} (N_z W_{dg})^{0.49} + S_w W_p \tag{4.12}$$

The response W denotes the wing weight (lb). This is a benchmark example for emulation, prediction, input-screening test problems [120]. The input variables and their calibration ranges are shown in Table 6.1. Among the input variables with intuitive names, the taper ratio, aspect ratio and the airfoil thickness to chord ratio are the terms related to the aircraft wing geometry. The quarter chord sweep defines the wing sweep angle between a perpendicular to the centerline and the leading edge of the wing. The ultimate load factor is an additional factor of safety to account for unexpected loads

above the design limits. Because this work focuses on developing the methodology of model calibration, the deep physics of the airfoil design is omitted.

Input variables' range	Input variables' meaning
$S_w \in [150, 200]$	Wing area (ft^2)
$W_{fw} \in [220, 300]$	Weight of fuel in the wing (lb)
$A \in [6, 10]$	Aspect ratio
$\Lambda \in [-10, 10]$	Quarter-chord sweep (degrees)
$q \in [16, 45]$	Dynamic pressure at cruise (lb/ft^2)
$\lambda \in [0.5, 1]$	Taper ratio
$t_c \in [0.08, 0.18]$	Airfoil thickness to chord ratio
$N_z \in [2.5, 6]$	Ultimate load factor
$W_{dg} \in [1700, 2500]$	Flight design gross weight (lb)
$W_p \in [0.025, 0.08]$	Paint weight (lb/ft^2)

TABLE 4.3: The input variables and their ranges for wing weight function

Model discrepancy is assumed to be 3. The observation is assumed to be $130lb$, with a standard deviation of 2. For each sequential wave, 20 new training data are added, chosen from the non-implausible input domain of the current wave. For Subset Simulation, the number of samples in each level strikes a balance between the computation speed and the coverage of the input space $n_L = 6000$ and the level probability strikes a balance between the required number of levels and efficient conditional sampling $p_L = 0.1$. HM was concluded when the emulator have a posterior variance smaller than the remaining uncertainties (OU and MD), which implies that the size of the non-implausible domain was unlikely to decrease in the next wave. Different sampling schemes with their adaptive versions (adaptively choosing the spread of the proposal PDF as the same order as the spread of the current sample's distribution in a level) are compared in Table 4.4. Their resulting non-implausible input domains are similar, shown by the upper triangle panel of Figure 4.11 using MMA. The lower triangle panel of Figure 4.11 shows optical depth plots [45]: on each plane of inputs' pairs, a 20×20 grid was created, with the color denoting the estimated probability of encountering a non-implausible sample over each grid. Optical depth plots give implausibility information perpendicular to projective input planes.

An advantage of using this wing weight model is that the physical meaning of the variables can help to understand the result of SSHM. From Equation 4.12, the range of the model output (wing weight) on the input space can be calculated, that is $(123, 511)lb$. The observation $130lb$ is a small output value on the input space. A small wing weigh

should have the physical configuration of a small wing area, a small weigh of the fuel in the wing, a small aspect ratio, a small dynamic pressure at cruise, a small taper ratio, a small ultimate load factor, a small flight design gross weight, a small paint weight, a large aerofoil thickness to chord ratio, and a moderate quarter-chord sweep. As can be seen from Figure 4.11, the resulting non-implausible samples are in line with this configuration.

MCMC algorithms	Distinct non-implausible samples	HM waves	Non-implausible probability	c.o.v.
MMA	904	8	8.17×10^{-6}	0.019-0.043
MMA (adaptive)	1835	8	7.95×10^{-6}	0.017-0.037
Repeated sample generation	329	7	6.67×10^{-6}	0.022-0.048
Repeated sample generation (adaptive)	1212	8	6.41×10^{-6}	0.018-0.039
Delayed rejection	447	7	7.74×10^{-6}	0.021-0.047
Delayed rejection (adaptive)	1434	7	7.57×10^{-6}	0.018-0.039
Subset-infinity	1098	7	7.88×10^{-6}	0.018-0.040
Adaptive optimal scaling	2771	8	9.90×10^{-6}	0.016-0.035

TABLE 4.4: The performance comparison among different MCMC sampling approaches, with parameters for Subset Simulation: $p_L = 0.1$ and $n_L = 6000$.

It is worth noticing that the number of distinct non-implausible samples (the distinct configurations of the non-implausible input parameters) varies greatly. One of HM's aim is to produce enough distinct non-implausible samples. However, MCMC algorithms may reject samples from the proposal movement, which produces repetitive sample points. Therefore this is an important criterion for comparing different MCMC algorithms applied in HM. The adaptive MCMC with optimal scaling obviously produced more distinct non-implausible samples. This is because the adaptive MCMC with optimal scaling choose the spread of the proposal PDF according to the optical acceptance probability, which maintains a relatively high acceptance rate of the sample candidates but also relatively widely explores the input space.

Comparing with the example in subsection 4.4.2, where Subset-infinity had more distinct non-implausible samples, both adaptive MCMC with optimal scaling and Subset-infinity cut out the first rejection event in MCMC to improve the acceptance rate, but Subset-infinity simply chose a more 'correct' spread of the proposal PDF $s = 0.01$ for that

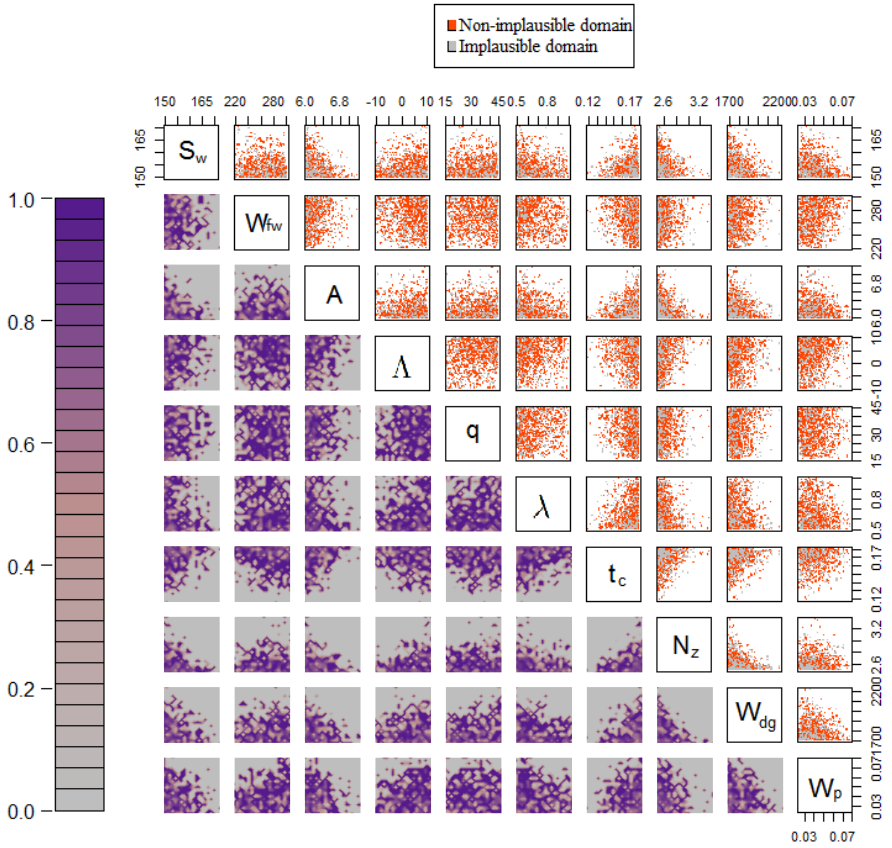


FIGURE 4.11: The upper triangle panel shows the implausibility of samples at the final-wave for the observations 130 *lb*. The orange area denotes the non-implausible domain (92% non-implausible samples out of all samples at the final-wave) and the gray area denotes the implausible domain. The lower triangle panel is the optical depth plot that shows the probability of encountering a non-implausible sample on 2-D projective grids of input pairs.

specific example. The other benchmarks are similar for all sampling schemes, e.g. all sampling schemes require 7 or 8 waves of SSHM.

Direct Monte Carlo (MC) is used to verify the result of SSHM. There are 5×10^7 samples randomly generated from the initial calibrating space. Each sample is evaluated by the wing weight model Equation 4.12. As a result, 253 samples can provide the match between the model output and the observation $130lb$. The model discrepancy is 3, the observation uncertainty is 4, while the code uncertainty does not exist any more because an emulator is not involved. The non-implausible probability evaluated by Direct MC is 5.06×10^{-6}, which is smaller than this value evaluated by SSHM. This is also because an emulator is not involved—the CU does not exist so that the implausibility measure for each sample becomes larger (Equation 5.1), hence more samples are determined as implausible. It can be comprehended as that the implausible threshold is tightened

without CU. Direct MC costs 5×10^7 samples to find 253 non-implausible samples, while SSHM only used 231000 or 264000[1]samples (with $240 - 260$ training samples) to find a lot more non-implausible samples (Table 4.4). The non-implausible samples are shown by Figure 4.12. As it can be seen that trend for the non-implausible domain is the same compared with SSHM. The optical dept plot is not produced because the total number of samples is too big and few non-implausible samples are invisible.

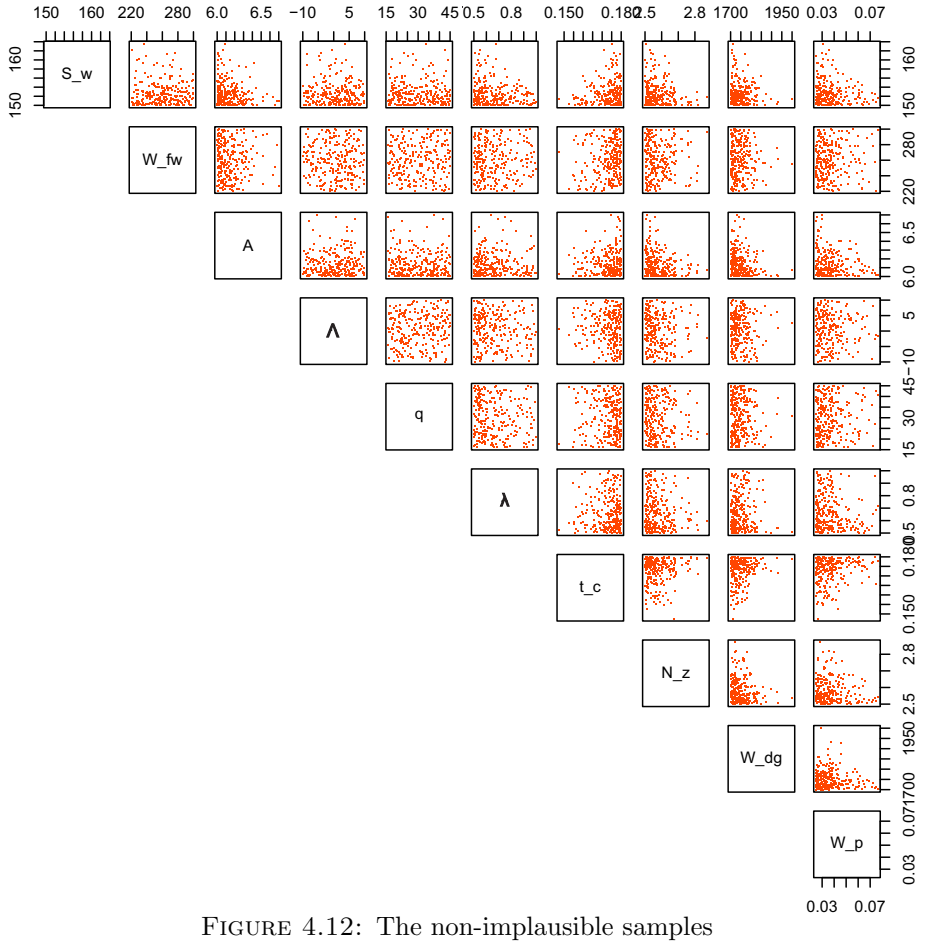


FIGURE 4.12: The non-implausible samples

[1]In each wave of SSHM, the number of samples generated by Subset Simulation is calculated by $n_L + L(n_L - n_L p_L)$, where L is the total number of levels, n_L is the number of samples in each level and p_L is the level probability. All MCMC sampling schemes considered in Table 4.4 take 7 or 8 waves of SSHM. In this case, the number of samples used by SSHM is $(6000 + 5(6000 - 6000 \times 0.1)) \times 7$ or 8, that is 231000 or 264000.

4.6 Classification for Training Data

4.6.1 DBSCAN

HM iteratively refocuses on the non-implausible domain. Instead of screening the input space to select new training data on the non-implausible domain, the non-implausible samples defined by the current wave can be reused as the new training data. The non-implausible domain can be disconnected or have a nontrivial topology. To ensure that the emulator is updated on every region of the non-implausible domain, a classification method should be applied to detect different regions, then new training data are generated on each of them separately. A documentation of a Python library [1] has listed the 9 widely used clustering methods (Figure 4.13), among which Density Based Spatial Clustering of Applications with Noise (DBSCAN) [53] is the most suitable for detecting disconnected domain. It is a density-based algorithm, which locates regions of high density that are separated from one another by regions of low density. Other algorithms either classify one 'connected' domain into multiple clusters or classify some disconnected domain into one cluster. The advantages of DBSCAN also include that it needs not to specify the number of clusters in data, it is robust to outliers via defining noise samples, it only needs two parameters, and it can classify arbitrarily shaped clusters with different sizes, even finding a cluster completely surrounded by another cluster. The concept and algorithm of DBSCAN are explained as follows.

DBSCAN needs two paramenters: ε and $minPts$. Sample points are classified as core points, border points and noise w.r.t. ε and $minPts$.

Definition 4.1 (Core point)**.** A point p is a core point if at least $minPts$ points are within distance ε of it.

Definition 4.2 (Border point)**.** A point q is a border point if there is a path $p_1, ..., p_n$ with $p_1 = p$ and $p_n = q$, where all p_i are core points.

In a word, core points are inside clusters, border points are on the border of clusters, and all other points are noise (outliers).

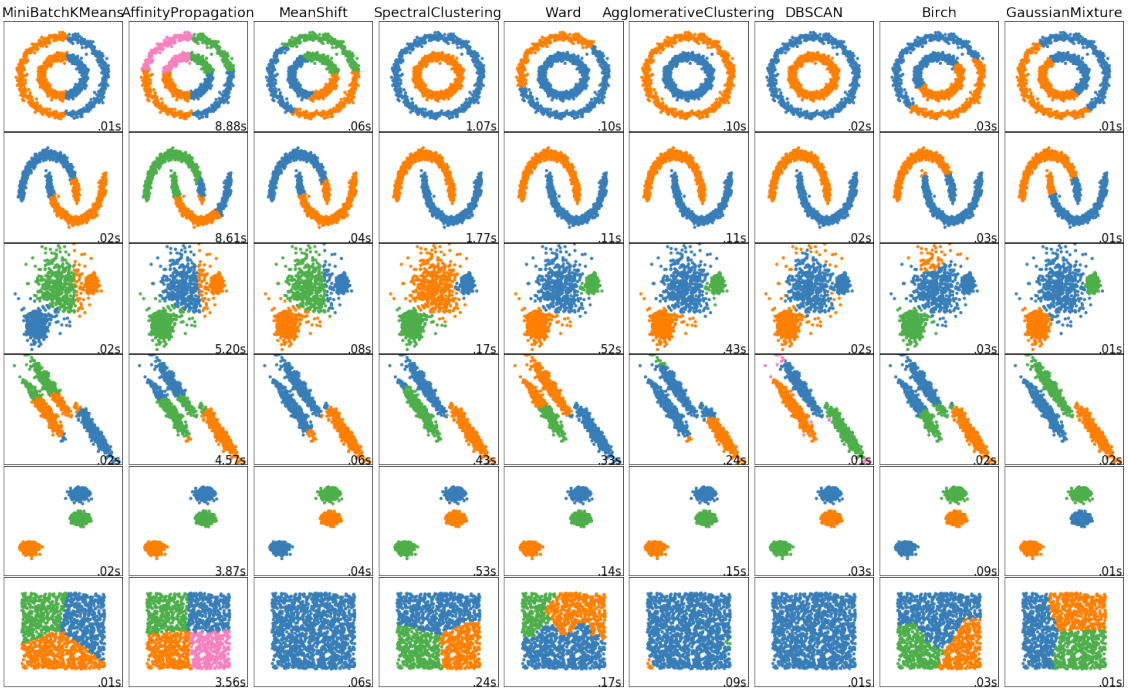


FIGURE 4.13: A comparison of the clustering algorithms from scikit-learn [1].

The classification algorithm of DBSCAN is as follows:

1. For an unvisited point, if there are at least $minPts$ points in its neighborhood within distance ε, all these points start a cluster and are considered visited;
2. Otherwise, the point is noise with the possible exception as a part of other clusters that will be discovered later;
3. Unvisited points in ε-neighborhood of the visited points are processed following steps 1 and 2, until all border points are found, then a cluster is complete;
4. Start with a new unvisited point to discover more clusters.

To determine the value of ε and $minPts$, an observation based heuristic [53] is used, trying to be robust against noise. It chooses the most number of samples belonged to clusters and others as noise. This heuristic orders each point by the distance from its kth nearest neighbor (k-NN distance). As shown by Figure 4.14 with $k = 4$, a threshold is taken at at the knee of the line, where a sudden increase appears in the number of points belonged to clusters. Hence, the parameters are determined as $\varepsilon = a$ and $minPts = 4$, and DBSCAN is then performed according to these two parameters. The number of

neighbors k considered in the heuristic figure is proposed to be fixed at $k = 4$ [53]. Heuristic graphs for $k > 4$ do not differ much from when $k = 4$, but they increase the computation amount.

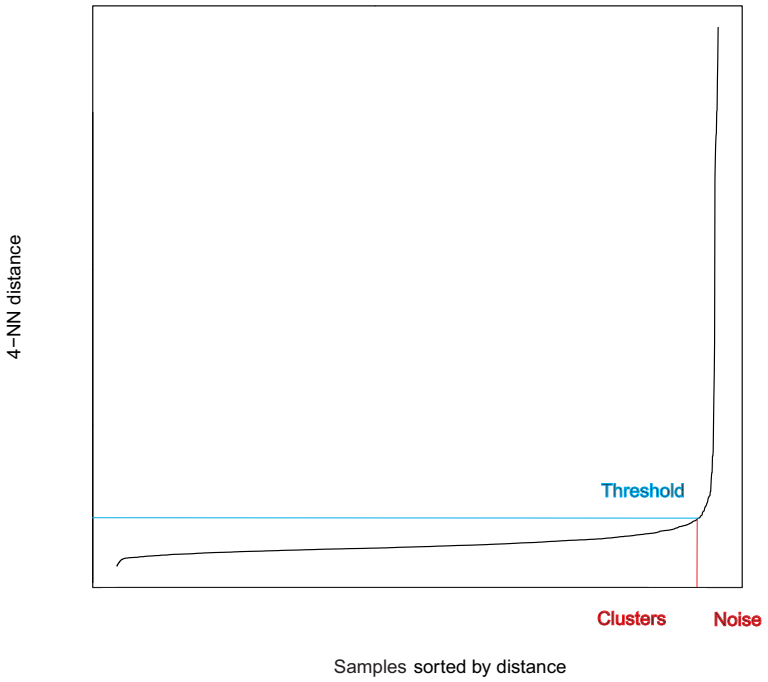


FIGURE 4.14: A heuristic to decide the number of clusters. The 4-NN distance determines the number of points in clusters. To detect the 'thinnest' cluster, the knee in the figure is a threshold, after which samples in clusters suddenly increase.

After the non-implausible input domain defined by the current wave is classified, the new training data for the next wave are resampled from the clusters respectively. The number of new training data selected in each cluster is proportional to the size of the cluster.

4.6.2 Numerical examples with multiple non-implausible regions

A numerical example that can visualize the advantage of sampling new training points from clusters is illustrated here. The model [121] (Figure 4.15) is chosen because it is known to be multi-modal, thus it is expected to generate disconnected domain:

$$f(x_1, x_2) = 3(1-x_1)^2 exp(-x_1^2-(x_2+1)^2) - 10(\frac{x_1}{5}-x_1^2-x_2^5)exp(-x_1^2-x_2^2) - \frac{1}{3}exp(-(x_1+1)^2-x_2^2) \tag{4.13}$$

The surface defined by this model is shown by Figure 4.15. The specified observation is assumed as 3 units, with the observation uncertainty $OU = 0.04$. There are 60 training samples at the first wave. After the first wave, all non-implausible points are classified as shown in Figure 4.17. The parameters $\varepsilon = 0.2$ and $minPts = 4$ are decided by the heuristic Figure 4.16. There are 7 clusters of non-implausible samples. Each cluster should have at least one new sample, thus 8 new training samples are added to the emulator, with the largest cluster has two new samples. After the second wave, the non-implausible samples have exceed 50% of all samples. A large amount of acceptable runs can be generated on the current non-implausible domain, thus SSHM can be concluded. The final non-implausible domain for $f(x_1, x_2) = 3$ is plotted in Figure 4.19.

Note that there are 7 non-implausible clusters defined by DBSCAN after the first wave, while the desired number of non-implausible clusters should be 3 observed from the surface of this model (Figure 4.15). This is because the emulator's precision is not high enough. After the refinement of the emulator at the second wave, if using DBSCAN to classify the final non-implausible domain, Figure 4.18 shows the resulted 3 clusters as expected.

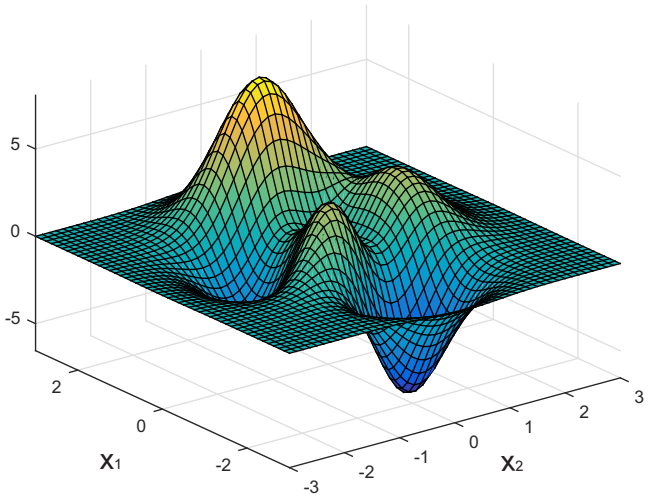


FIGURE 4.15: Surface of the model output.

This example has visually shown that DBSCAN can classify samples for disconnected non-implausible domain of SSHM. New training data can then be selected from each cluster to improve the accuracy of the emulator on each region of the disconnected

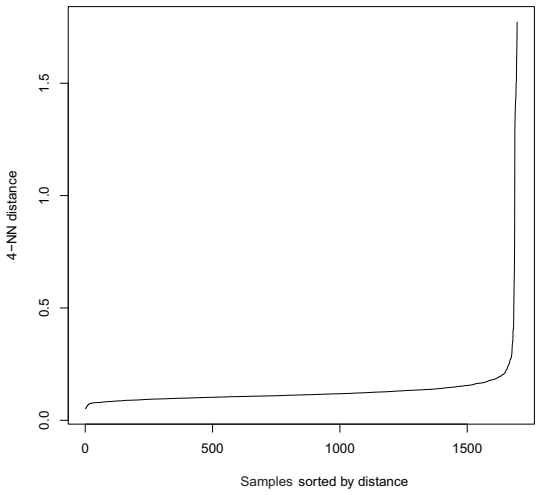


FIGURE 4.16: Observation based heuristic algorithm to decide parameters for DBSCAN: $\varepsilon = 0.2$ and $minPts = 4$.

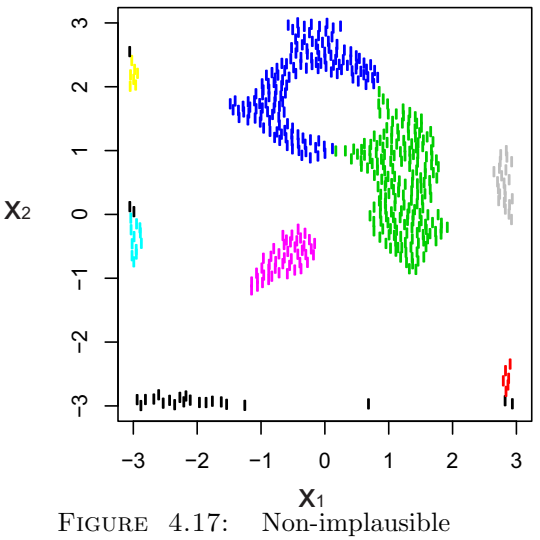


FIGURE 4.17: Non-implausible samples after the first wave. There are 7 clusters in different colors. Black points represent noise.

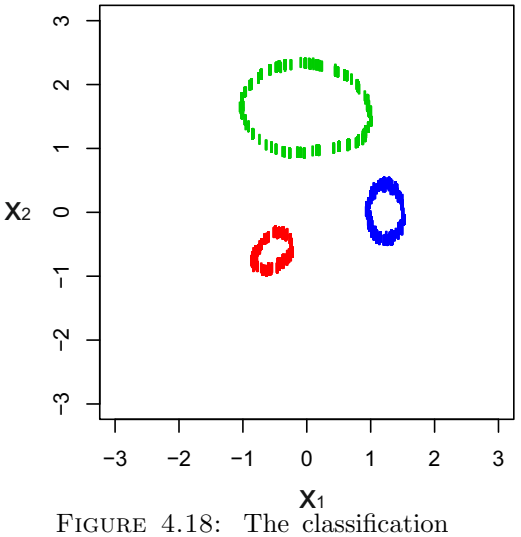


FIGURE 4.18: The classification by DBSCAN for the resulting non-implausible domain.

non-implausible domain in every wave. Otherwise, if new training data are only generated in part of regions of the disconnected non-implausible domain, the resulting non-implausible input domain may end up to as Figure 4.20, where some samples in the left-out regions cannot provide an accurate match. Although this can be circumvented by using the stopping criterion of attaining an accurate emulator, HM may cost more waves until the emulator at all non-implausible regions is refined.

For the 10-D wing weight model, if two observations 130 *lb* and 460 *lb* are assumed, it can have two disconnected con-implausible regions. This is shown by Figure 4.21 with

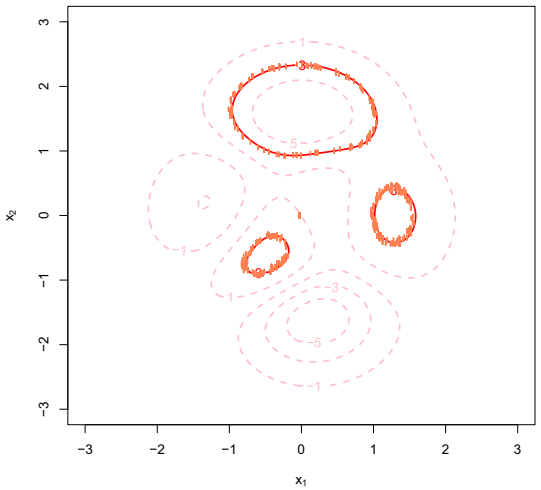


FIGURE 4.19: Non-implausible input domain denoted by orange dots, based on new training data for the second wave generated using DBSCAN.

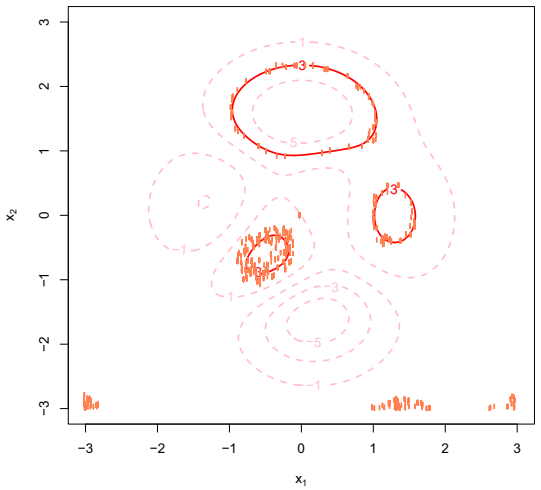


FIGURE 4.20: Non-implausible input domain denoted by orange dots, based on new training data for the second wave only generated from two regions of the disconnected non-implausible domain defined by the first wave.

the non-implausible samples and the optical depth plots. This result is obtained by SSHM using DBSCAN. The parameters for SSHM were the same as when there was only one observation. The heuristic plot for every wave is similar, shown by Figure 4.22. The parameters for DBSCAN were chosen as $\varepsilon = 30$ and $minPts = 4$ and DBSCAN successfully classified two disconnected non-implausible regions at each wave.

4.7 Summary

In this chapter, the implementation of Subset Simulation for non-implausible sampling was introduced. Non-implausible domains can be difficult to sample because of the non-trivial topology, the disconnected property or being only a small fraction of the design space. However HM demands a sampling method to screen the input space in each wave to refocus on the non-implausible domain, which causes computational issues for implementing HM. We proposed to apply Subset Simulation to generate samples in such situations. The potential was shown by the numerical examples for sampling w.r.t. implausibility functions. Numerical examples implementing SSHM were then illustrated. First, a low dimensional example visually showed the implementation process of SSHM, especially the refocusing: resampling on the non-implausible domain, evaluating new

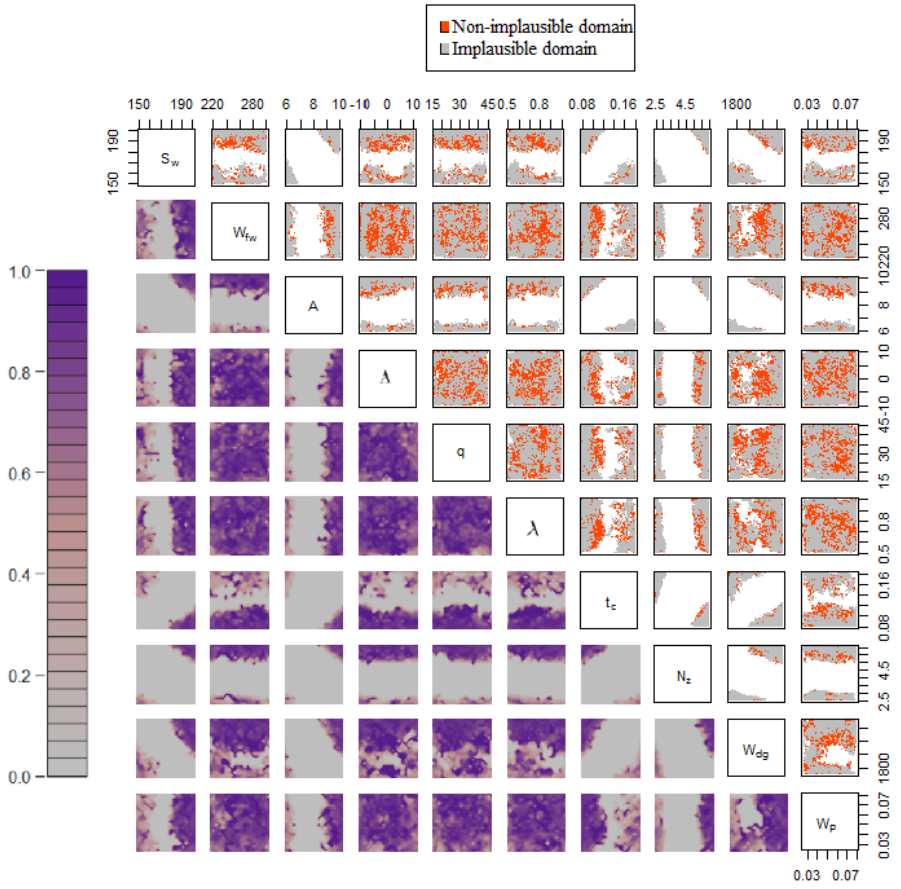


FIGURE 4.21: The upper triangle panel shows the implausibility of samples at the final-wave for observations 130 *lb* and 460 *lb*. The orange area denotes the non-implausible domain (81% non-implausible samples out of all samples at the final-wave) and the gray area denotes the implausible domain. The lower triangle panel is the optical depth plot that shows the probability of encountering a non-implausible sample on 2-D projective grids of input pairs.

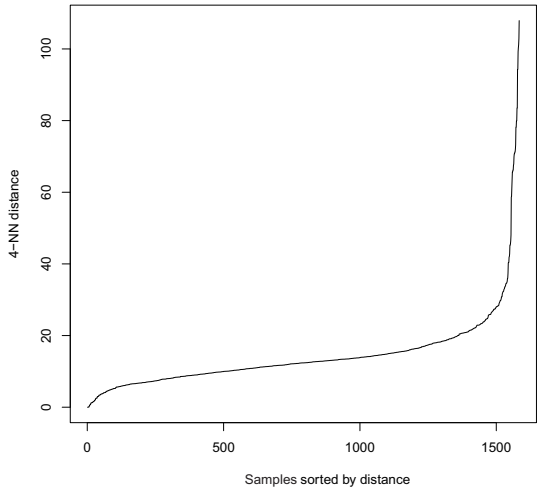


FIGURE 4.22: The heuristic for after the first wave to decide parameters for DBSCAN. In this case, $\varepsilon = 30$ and $minPts = 4$ [2].

training data using the simulator, refining the emulator and testing the implausibility of the samples using the output of the refined emulator. Second, a 10-D example drew some

conclusions through comparing different variants of sampling schemes in Subset Simulation: adaptive variance commonly gave more distinct combinations of input parameters; the efficiency of Subset-infinity depended on the spread of the proposal distribution; and the adaptive MCMC with optimal scaling produced more distinct non-implausible samples because it maintained the optimal acceptance rate for sample candidates.

A strategy to generate new training data was proposed. It reuses the non-implausible samples in the current wave. In order to have the training samples cover each disconnected region of the non-implausible domain, a classification method—DBSCAN was applied. New training samples are hence generated on each cluster detected by DBSCAN. A numerical example with multiple non-implausible regions presented the implementation of this proposed method. It showed that classifying training data can improve the accuracy of an emulator, so that improved the accuracy of the resulting fitting input.

The proposed SSHM had shown to be applicable for non-implausible sampling in HM via these above numerical examples. The next step is to apply SSHM to solve realistic engineering problems. An Industrial application will be illustrated in the following chapter.

[2]Note that although there were 6000 samples for each wave, after MCMC repetitive samples were generated. For DBSCAN only unique samples were clustered. In this wave there were 447 unique samples, so that there were 1788 points sorted by their 4-NN distance.

Chapter 5

Industrial Application: Climb-Cruise Engine Matching

5.1 Introduction

Once all the theoretical results have been established, SSHM is applied to solve a real-world industrial challenge—the'climb-cruise engine matching' problem, which was posed by Airbus at the 'Uncertainty Quantification and Management (UQ&M)' study group, hosted by the Institute for Risk and Uncertainty at the University of Liverpool, July 2017. Industry partners were invited to bring challenges in the fields of mathematics, statistics, engineering and computer science, which were then studied by about fifty academics invited from the UK, in an structured and intense session over three days, involving discussing theoretical solutions, trying to solve problems and presenting results. This work provides a solution to the 'climb-cruise engine matching' problem posed by Airbus.

The 'climb-cruise engine matching' problem aimed to narrow the set of feasible aircraft configurations so that can provide competitive advantages in the modern aviation market. 'Flightpath 2050' [122] lunched by the high level group on aviation research of the European Commission has pointed out that the industrial competition of the modern aviation market for European aviation has become ever fiercer from notable challengers

such as the US, Brazil, Canada, China, India and Russia. Because the effect of aviation on the atmosphere is understood, modern aircraft are expected to operate under stricter restrictions in order to protect the environment and the energy supply. To meet this societal/market need, it is a great challenge to seek feasible aircraft configurations making a friendly environment innovation.

A robust design of the airframe and engine is desired, which deals with multiple targets with uncertainty tolerances, targets such as safety and security of the passengers and freight, transport speed, as well as less negative effects on the environment. The overall concept of the aircraft design approach is shown by Figure 5.1. The 'climb-cruise engine

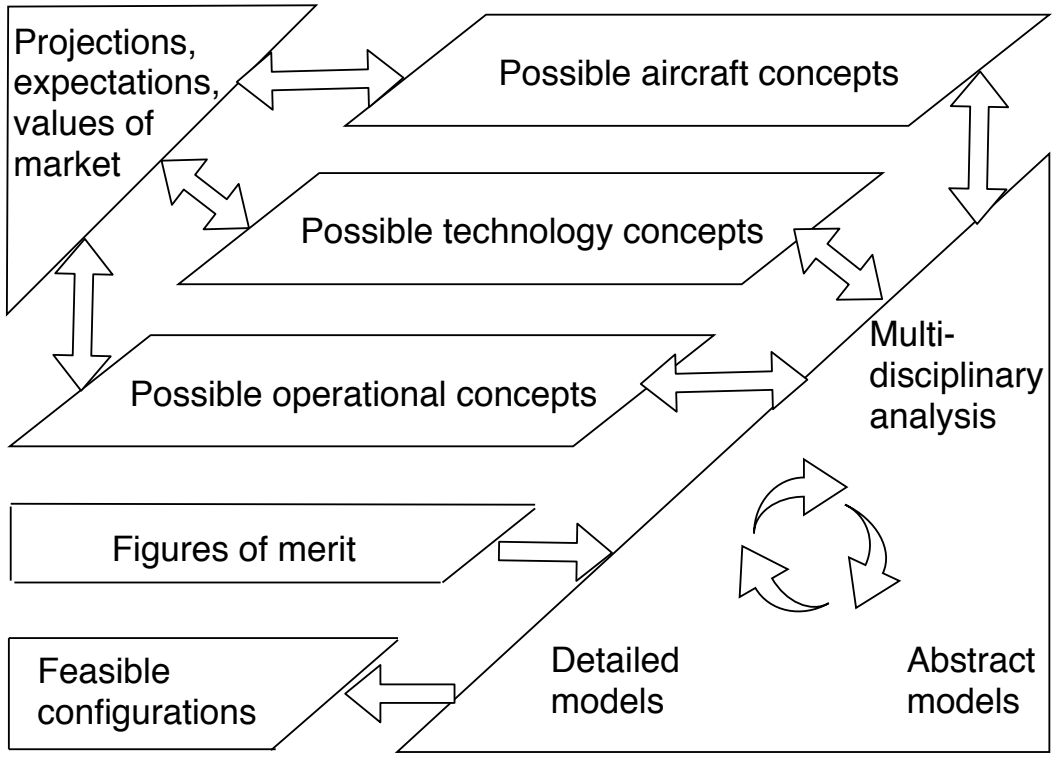


FIGURE 5.1: The overall concept of the aircraft design approach.

matching' is a robust design problem, which tries to find appropriate airframe and engine parameters aiming to make less noise, less emission and use less fuel. The corresponding physical quantities of interest are the flyover noise, the sideline noise, the Nitrogen Oxide emissions, and the block fuel, which are called the figures of merit. They are used to characterize the performance of the aircraft to find the fitting configurations from a multitude of airframes and engines, listed in the following table.

Figures of merit	Unit	Attribute
Flyover noise	dB	Environmental impact
Sideline noise	dB	Environmental impact
Nitrogen Oxide emissions	lb	Environmental impact
Block fuel	lb	Performance efficiency

TABLE 5.1: The figures of merit for 'climb-cruise engine matching'.

For the design parameters, the wing area and the wing aspect ratio are considered to create the configuration of wings. The see level static thrust, the fan pressure ratio, the overall pressure ratio, the bypass ratio and the fuel flow factor are considered to create the configuration of engines. The overall drag and the empty weight are considered to create the configuration of the overall airframe. Because this work focuses on developing the methodology of model calibration, the deep physics of airframe design is omitted.

Design parameters	Unit	Attribute
Wing area	ft^2	Wing
Wing aspect ratio	/	Wing
Sea level static thrust	lbf	Engine
Fan pressure ratio	/	Engine
Overall pressure ratio	/	Engine
Bypass ratio	/	Engine
Fuel flow factor	/	Engine
Overall drag	lbf	Airframe
Empty weight	lb	Airframe

TABLE 5.2: The design parameters for 'climb-cruise engine matching'.

To perform the robust design, simulators were built to model a 24-hour operation aircraft by the AirCADia framework [123], which was conceived and developed by the Advanced Engineering Design Group at Cranfield University as part of several industry-led research projects (e.g. CONGA, TOICA [123, 124]). The AirCADia framework consists of hundreds of plug-in models for analyzing various aircraft designs. The conceptual schematic of the models used in this climb-cruise engine matching problem is shown by Figure 5.2. The gaseous emissions model and the jet noise model are coupled with the climb performance model and the cruise performance model to compute the flyover noise, sideline noise, Nitrogen Oxide emissions and the block fuel during the climbing and cruising performance of an aircraft.

In order to solve the robust design problem, it was proposed to draw an analogy with HM. If the targets with uncertainty tolerances are treated as observations with measurement uncertainty, this robust design problem can be viewed as a calibration problem

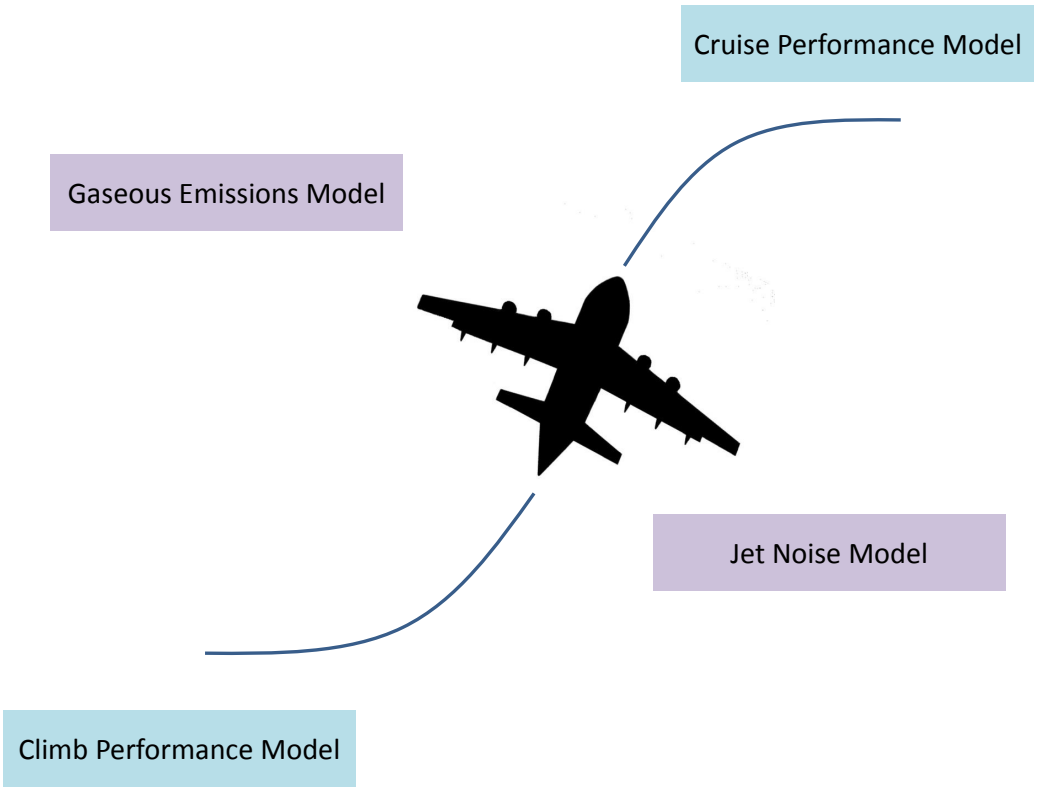


FIGURE 5.2: The conceptual schematic of the models used in the climb-cruise engine matching problem.

for multi-dimensional and complex simulators. In this context, SSHM can be used to find the fitting input domain (parameters) that can provide a match between targets (observations) and model outputs. Vast volume of input space that cannot reproduce the observation can be cut off efficiently by the refocusing process of SSHM. The use of an iterative succession of emulators makes inference for complex simulators, thus reduces the computation time. The implementation and the result are presented as follows.

5.2 Climb-cruise engine matching

Due to the analogy between robust design and calibration, targets with uncertainty tolerances was proposed to be treated as observations with measurement uncertainty. Hence this problem can be solved in the framework of SSHM to find the fitting parameters that provide a match between model outputs and observations (targets). Based on

the objectives (noise curfews and controls of fuel consumption and gaseous emissions), the figures of merit (model outputs) are the flyover noise, the sideline noise, the Nitrogen Oxide emissions and the block fuel. Their values should all be the lower the better, but considering limitations of the aviation technologies, targets are set at some specified values, regarded as observations. The AirCADia models were calibrated to find the fitting input parameters out of the initial input domain, which is a set of representative single aisle aircraft configurations of a multitude of airframes with a multitude of engines. The description and initial ranges for the model inputs are shown by Table 5.3. The initial ranges of the model inputs are decided by wealth of domain knowledge. For confidential reasons in business, the design targets, or observations are realistic but not real. The design target was set after consultation with Airbus as 83 dB for the flyover noise, 86 dB for the sideline noise, 270 lb for the Nitrogen Oxide emissions, and 30000 lb for the block fuel. The uncertainty tolerances or observation uncertainties (OU) and the model discrepancies were assumed as 1% of the target values for the four outputs separately. SSHM can be used to find the fitting parameters that provide a match between model outputs and these values. Because multiple outputs $g_j(\mathbf{x})$ $j = 1, ..., r$ were calibrated, the fitting input domain should satisfy $\frac{|z_1 - E^*(g_1(\mathbf{x}))|}{V_1(\mathbf{x})} \leq 3$, $\frac{|z_2 - E^*(g_2(\mathbf{x}))|}{V_2(\mathbf{x})} \leq 3, ..., \frac{|z_r - E^*(g_r(\mathbf{x}))|}{V_r(\mathbf{x})} \leq 3, r = 4$ simultaneously. Hence, the implausibility threshold for multiple outputs was as follows:

$$I(\mathbf{x}) = max(I_j(\mathbf{x})), j = 1, ..., r \tag{5.1}$$

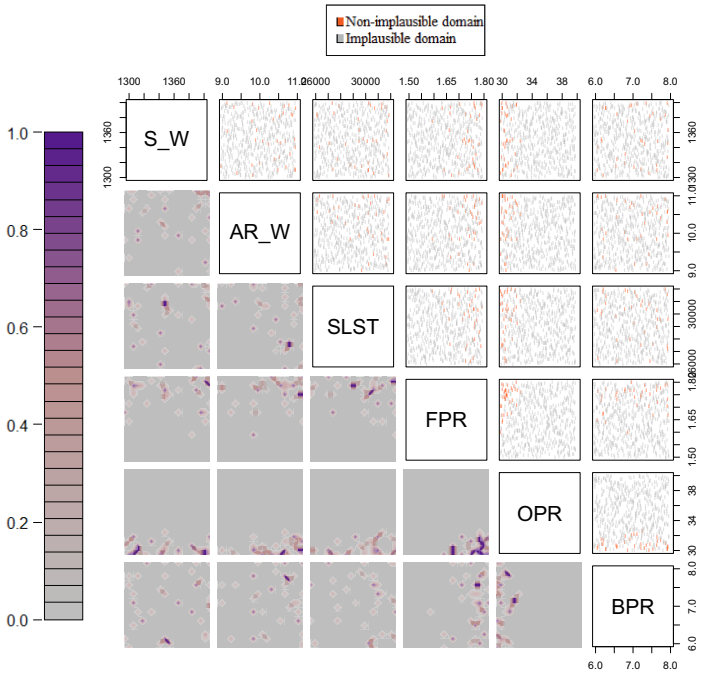
There were 90 initial training samples, 5000 emulation points and 40 new training samples in each wave.

Input variables' range	Input variables
$S_W \in [1300, 1400]$	Wing area (ft^2)
$AR_W \in [9, 11]$	Wing aspect ratio
$SLST \in [26000, 32000]$	Sea level static thrust (lb)
$FPR \in [1.5, 1.8]$	Fan pressure ratio
$OPR \in [30, 40]$	Overall pressure ratio
$BPR \in [6, 8]$	Bypass ratio
$FFF \in [1, 1.2]$	Fuel flow factor
$OR \in [1, 1.2]$	Overall drag (lb)
$EW \in [0.01, 0.03]$	Empty weight (lb)

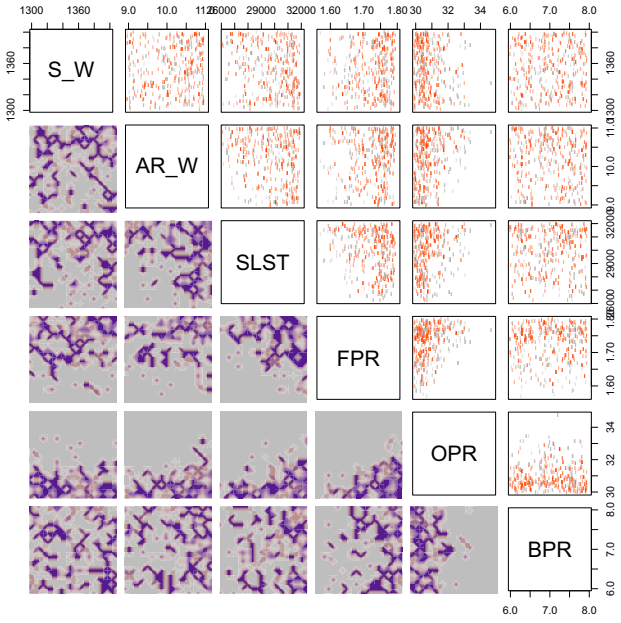
TABLE 5.3: The input variables and their ranges for 'climb-cruise engine matching'.

5.3 Analysis using SSHM

As a preliminary study, six inputs (the wing area, the wing aspect ratio, the sea level static thrust, the fan pressure ratio, the overall pressure ratio and the bypass ratio) and



(b)Wave1



(b)Wave2

FIGURE 5.3: Non-implausible and implausible samples found by SSHM for the target $240lb$ of the Nitrogen Oxide emissions.

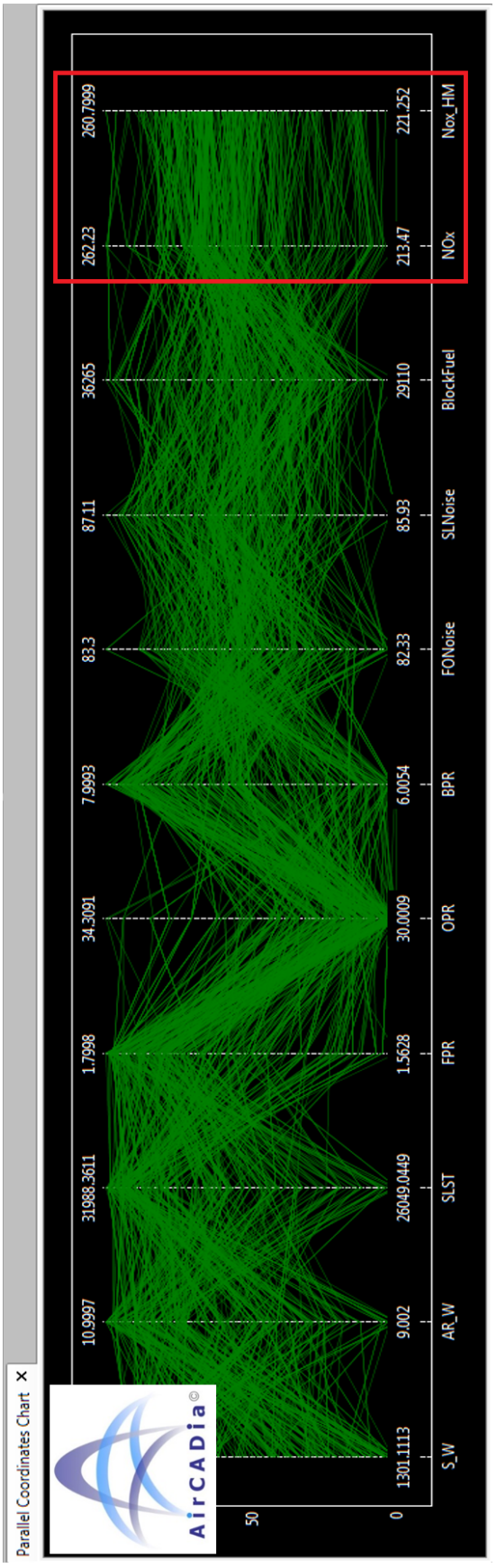


FIGURE 5.4: Parallel coordinates plot provided by AirCADia. The first 6 columns are the fitting non-implausible input configurations to the target 240*b* considering the uncertainties (OU, MD and CU). The next 4 columns are the model outputs computed by AirCADia. The last column shows the emulator's output of SSHM.

one output (the Nitrogen Oxide emissions) were considered. The design target was set after consultation with Airbus as $240lb$ for the Nitrogen Oxide emissions. The uncertainty tolerance or observation uncertainty (OU) was assumed as 2. The model discrepancy (MD) was assumed as 3. There were 60 initial training samples, 600 emulation points and 8 new training samples in each wave.

For this preliminary study, if a large number of acceptable runs can be generated from a region, SSHM can be concluded, which means that a fitting input domain that can reproduce given observations has been found. Two waves of SSHM were necessary. At the second wave, more than 50% samples were non-implausible. The results were

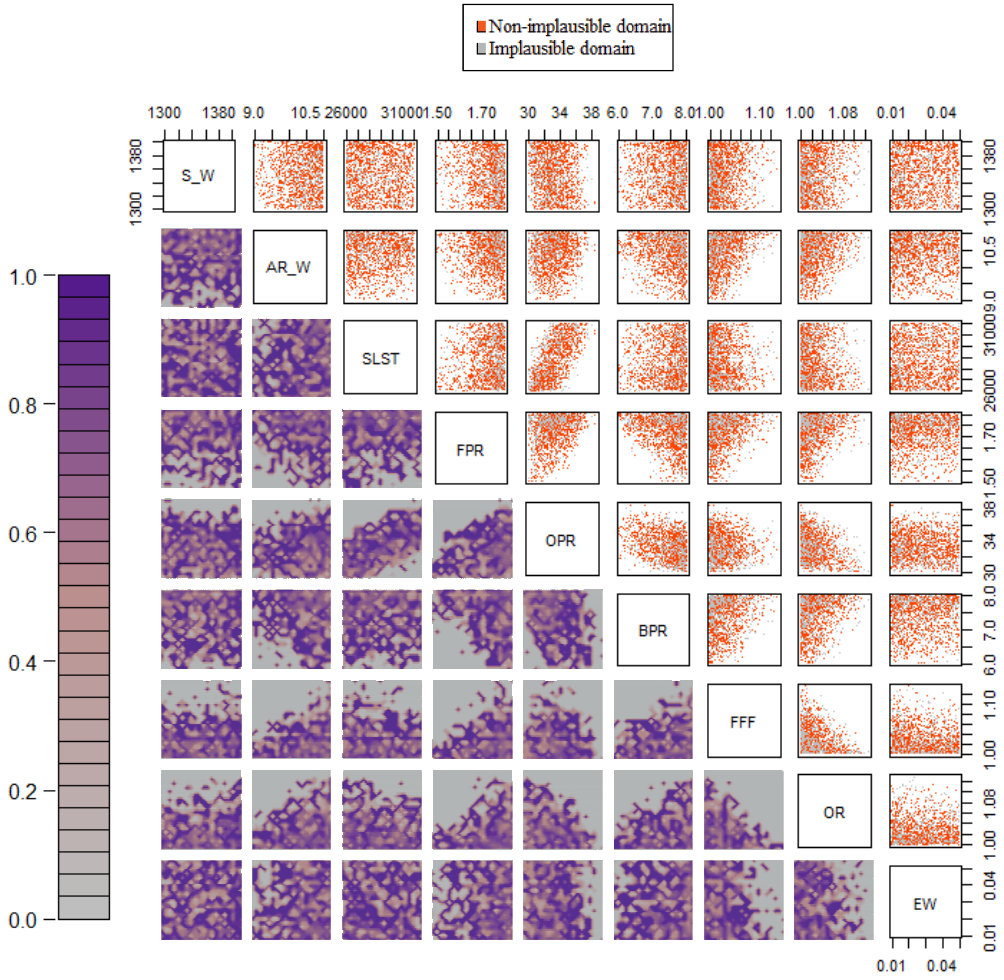


FIGURE 5.5: The upper triangle panel shows the final-wave non-implausible domain for targets: $83dB$ for the flyover noise, $86dB$ for the sideline noise, $270lb$ for the Nitrogen Oxide emissions and $30000lb$ for the block fuel. The lower triangle panel is the optical depth plot that shows the probability of encountering a non-implausible sample on 2-D projective grids.

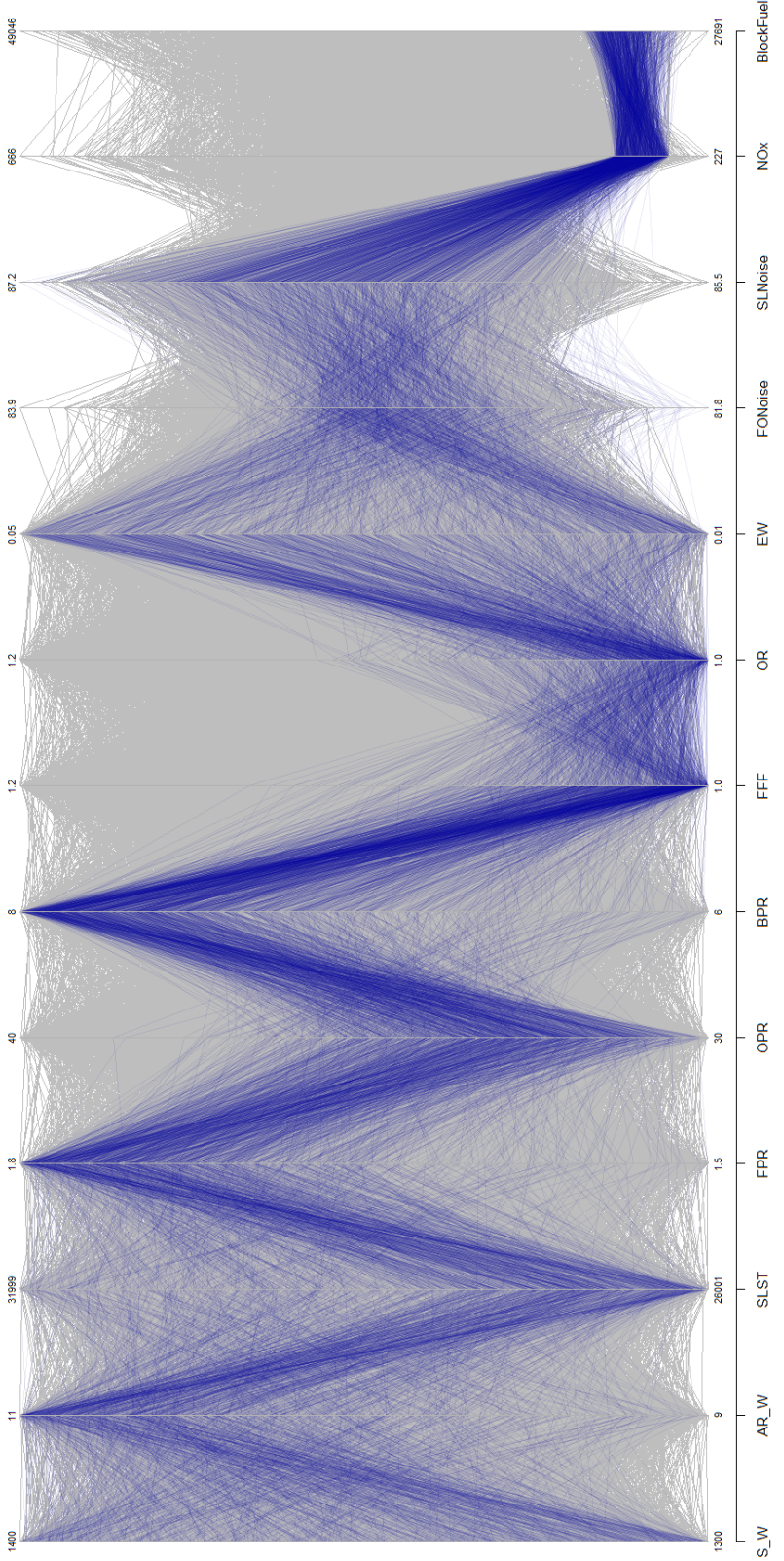


FIGURE 5.6: Parallel coordinates plot. The first 9 columns are the fitting non-implausible input configurations to the targets: $83dB$ for the flyover noise, $86dB$ for the sideline noise, $270lb$ for the Nitrogen Oxide emissions and $30000lb$ for the block fuel. The last 4 columns are the corresponding model outputs. The implausibility threshold is 3. The background color shows the design space and the blue lines show the non-implausible samples after HM.

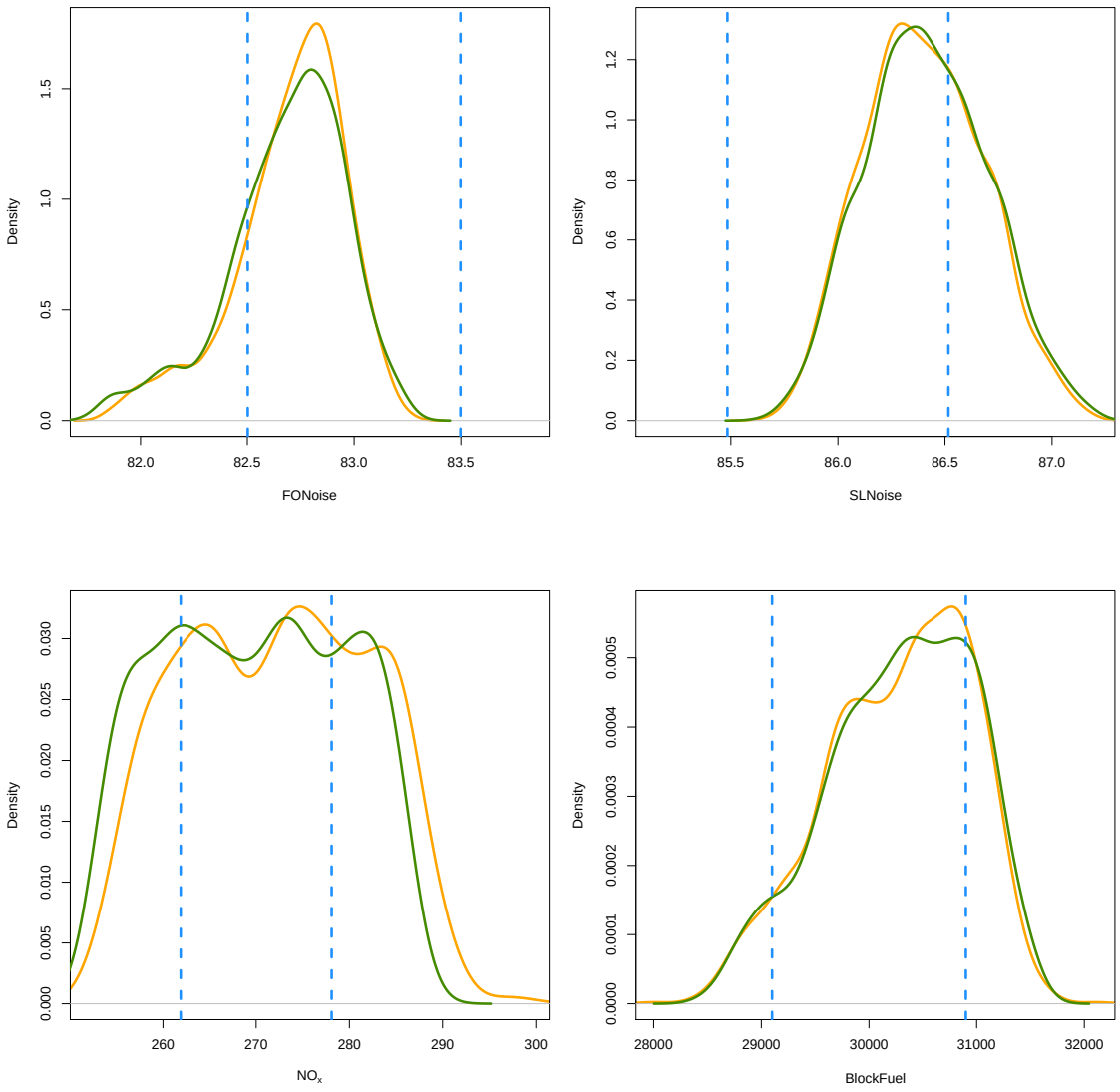


FIGURE 5.7: The kernel density for the output of the non-implausible samples, with the green line representing the emulator's output and the orange line representing the simulator's output. The target figures-of-merit with plus and minus the observation uncertainty are denoted by the dotted blue lines.

presented by Figure 5.3. The parallel coordinates plot (Figure 5.4) shows the fitting input domain together with the emulator's output, compared with the output provided by AirCADia. The parallel lines inside the red frame verify that the emulator's output is consistent with the simulator's output and the resulting input configurations can provide the desired match to the target $240lb$ considering the uncertainties (OU, MD and CU).

Based on that the success of the preliminary trial, SSHM was used for calibrating the simulators considering all inputs and outputs. SSHM concluded when the emulator's posterior variance (CU) is smaller than the remaining uncertainties (OU and MD). This

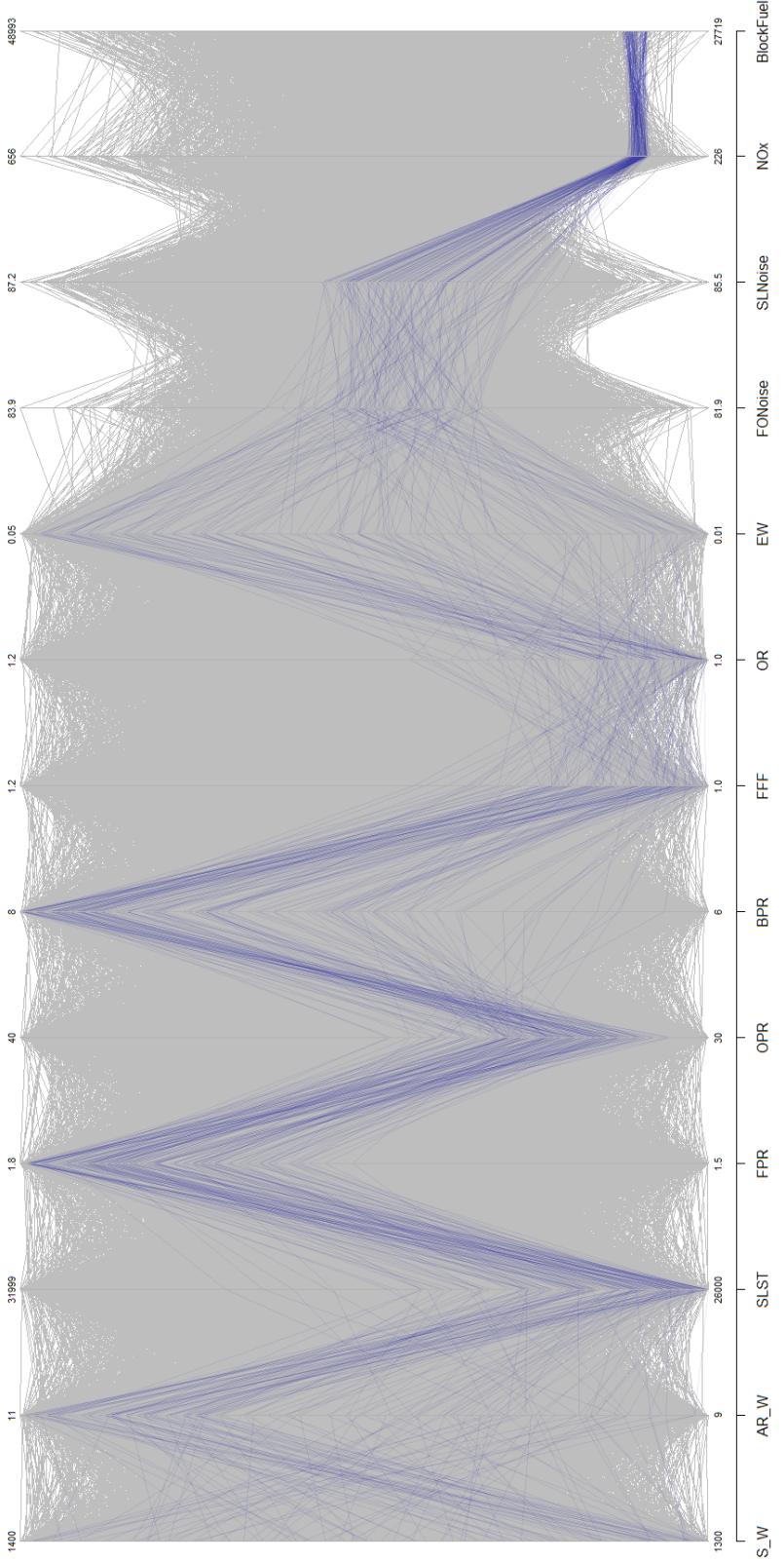


FIGURE 5.8: Parallel coordinates plot. The first 9 columns are the fitting non-implausible input configurations to the targets: $83dB$ for the flyover noise, $86dB$ for the sideline noise, $270lb$ for the Nitrogen Oxide emissions and $30000lb$ for the block fuel. The last 4 columns are the corresponding model outputs. The implausibility threshold is 1. The background color shows the design space and the blue lines show the non-implausible samples after HM.

implied that the non-implausible domain is unlikely to decrease in further waves. After 21 waves of SSHM, the matching input domain for the targets were found as shown in Figure 5.5. The upper triangle panel shows the scatter plots of all non-implausible samples in the last wave and the lower triangle panel shows the probability of encountering non-implausible samples. The parallel coordinates plot (Figure 5.6) shows the fitting input domain in another way, where resulting configurations of the model inputs (the first 9 columns) and outputs (the last 4 columns) can be presented together. The gray color in the background shows the wide possible input ranges before the calibration and the blue lines on the front show the non-implausible samples after the calibration. As can be seen, low values of the wing area, the sea level static thrust, the overall pressure ratio, the fuel flow factor and the overall drag, and high values of the wing aspect ratio, the fan pressure ratio, and the bypass ratio are inclined to provide a match to the specified observations.

It was shown that SSHM successfully narrows down the input design space given the specified observations. Figure 5.7 compares the kernel density [125] of the emulator's output with the AirCADia models' output. The two output lines (orange and green) generally coincide with each other, which verifies that the correct input space has been targeted by SSHM.

Because the non-implausible samples were defined by 3 times standard deviation of not just the OU but also the CU and MD, the non-implausible domain was wider than the blue window. In the first figure, e.g. if many fitting samples were found providing output around 82.8 (as long as surviving the implausibility threshold), the non-implausible domain could be centered around that value and therefore skewed.

As previously mentioned in section 2.6, if the implausibility threshold was changed to 1 instead of 3, only samples with their output lying within the domain around the observation with one standard deviation are retained. SSHM concluded when the emulator's posterior variance (CU) is smaller than the remaining uncertainties (OU and MD). After 16 waves of SSHM, the matching input domain for the targets were found as shown in Figure 5.8. The non-implausible probability is 2.3×10^{-4} estimated by Subset Simulation. Subset Simulation had proven to be robust for estimating small probabilities

($p_F \leq 10^{-3}$) [52]. The emulators' outputs have been benchmarked by the kernel density plot (Figure 5.7). Thus, this result was not verified by direct Monte Carlo, which would need running the simulator hundreds of thousands of times. SSHM used fewer waves with the implausibility threshold equal to 1 to reach the desired accuracy. This is because the model output is expected to behave smoothly on a smaller non-implausible space so that the behavior of the model output is easier to catch for the emulator.

5.4 Summary

In this chapter, the proposed SSHM was used to find aircraft configurations from a multitude of airframes and a multitude of engines to provide a match between model output and specific noise or gaseous emission level. Finding appropriate parameters in this problem was originally a robust design problem. In order to solve the robust design problem, it was proposed to draw an analogy with HM. The targets of design were interpreted as observations in calibration. Four independent figures-of-merit were considered at the same time in this application, reflecting in that the implausibility threshold considers four criteria at the same time. The usage of emulators was time-saving and confirmed feasible by comparing their outputs with the AirCADia's outputs. SSHM was proved efficient in exploring and reducing input parameter space with the given observations even though the target space of the above example is a tiny fraction compared to the initial input space.

Chapter 6

Expert Estimates and Correlation Structure

6.1 Introduction

This chapter expands the theory developed in previous chapters to explore the correlation between resulting input parameters and observations. New algorithms are proposed to gauge the uncertainty of expert estimated correlations. They can be used more generally than within the framework of HM.

In many practical situations, it is important to know the correlation between different variables of interest. For example, if oil locations are strongly correlated with certain types of geological structures, then oil should be actively looked for in an area where such structures are present. Similarly, if a certain disease is strongly correlated with e.g. smoking, then probably smoking weakens the body's natural defenses against this disease. In an ideal world, correlations should be determined from experiments. When data are difficult to collect from experiment measurements or service inspections, their correlations cannot be calculated due to not enough data. For example, in HM, there can be only a few observations or even only one observation, hence statistically calculating the correlation between the resulting non-implausible input and the observation is impossible.

A natural idea is to ask experts. However, expert estimates can be crude, resulting in a negative definite correlation matrix. A correlation matrix describes the dependency between multiple variables and should be a positive semi-definite matrix, which will be shown in section 6.2. The physical meaning of this conflict is that the correlations for different variables should be interrelated with each other. For example, calculating the correlation between variables x and y and the correlation between variables x and z are both affected by the samples of variable x. While experts fill in the correlation matrix with estimated correlation coefficients for each pair of variables separately. The estimated correlation coefficients only construct an approximate correlation matrix and lack of consistency. The positive definiteness of correlation matrices has been discussed in optimization algorithms, construction of linear regression models, and a wide variety of applications [126–128]. It is of both theoretical and computational importance. This chapter discusses how to quantify the uncertainty of the expert-estimated correlation matrix, so that a positive semi-definite correlation matrix can be obtained given expert estimates.

When experts are not certain about the correlation coefficient, they may provide multiple values in some set. To model this imprecision, fuzzy sets [51, 129, 130] provide interval estimates with uncertainty levels in the values included. They relax the need for precise estimates and allow the specification for uncertainty of the estimates. The degree of uncertainty is marked by the experts with a number on the scale from 0 to 1. The value 0 means the experts are certain, while 1 means the experts are not certain. The fuzzy sets are a family of intervals labeled by different degrees of uncertainty, in a nested fashion. This chapter tries to find the tightest intervals of the expert-estimated correlation coefficients, where there exist a positive semi-definite matrix.

6.2 Formulation of the problem

Let $\mathbf{A}$ denote the correlation matrix that quantify the linear dependency between different quantities $x_1, ..., x_n$, which consists of correlation coefficients a_{ij}, $i, j = 1, ..., n$ that

quantify the dependency between each pair of variables:

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \tag{6.1}$$

The correlation coefficient a_{ij} is calculated by:

$$a_{ij} = \frac{Cov(x_i, x_j)}{\sigma_{x_i}\sigma_{x_j}} = \frac{E((x_i - \mu_{x_i})(x_j - \mu_{x_j}))}{\sigma_{x_i}\sigma_{x_j}} \tag{6.2}$$

where $Cov(\cdot)$ is the covariance of x_i and x_j; $E(\cdot)$ represents expectation; μ_{x_i} and μ_{x_j} are the means of x_i and x_j; σ_{x_i} and σ_{x_j} are the standard deviations of x_i and x_j. The correlation between a variable and itself is 1, hence all diagonal ($i = j$) elements in the above matrix are equal to 1. The correlation between x_i and y_i is the same as the correlation between y_i and x_i, hence the correlation matrix is symmetric.

An ideal way to estimate the correlation between two quantities x and y is to have a sample of data points in which only x changes—all other parameters remain constant—and detect how y changes depending on the change in x. However, often there are not enough empirical data to determine all possible correlations. For example in aircraft engineering, the ideal situation cannot be expected (e.g. shown by Figure 6.1 according to the 'climb-cruise engine matching' problem in chapter 5), in which all the airframe characteristics are identical except for one feature. In most practical situations, there are many factors affecting each situation, and there are often not enough data points to separate the effect of these factors. A natural idea to circumvent this lack of data is to rely on expert judgement.

Experts can provide estimates for the correlation coefficient a_{ij} between different variables. However, expert estimates can only be approximate.. They may violate the fact that any correlation matrix, by construction, must be positive semi-definite. This fact is shown as follows. A symmetric $n \times n$ positive semi-definite matrix $\mathbf{A}$ means that for any vector $\mathbf{z} \in \mathbb{R}^n$, the scalar $\mathbf{z}^T\mathbf{A}\mathbf{z}$ is always non-negative, i.e., $\sum_{i,j} a_{ij} \cdot z_i \cdot z_j \geq 0$ is true.

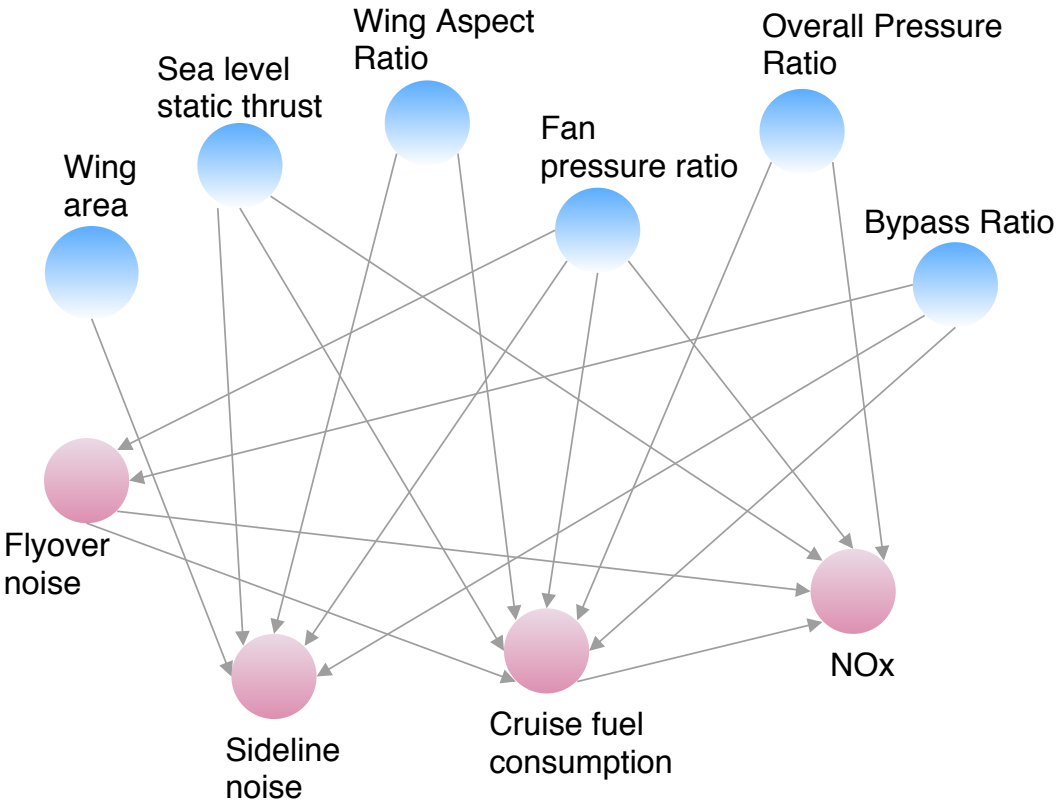


FIGURE 6.1: The complex dependency between variables for the industrial application: 'climb-cruise engine matching'.

According to Equation 6.2, the correlation coefficient a_{ij} can be rewritten as:

$$a_{ij} \stackrel{\text{def}}{=} \frac{E[\Delta x_i \cdot \Delta x_j]}{\sigma_i \cdot \sigma_j} \tag{6.3}$$

where $\Delta x_i \stackrel{\text{def}}{=} x_i - E[x_i]$ and $\sigma_i^2 \stackrel{\text{def}}{=} E[(\Delta x_i)^2]$. Thus, the desired sum

$$S \stackrel{\text{def}}{=} \sum_{i,j} a_{ij} \cdot z_i \cdot z_j = \sum_{i,j} \frac{E[\Delta x_i \cdot \Delta x_j]}{\sigma_i \cdot \sigma_j} \cdot z_i \cdot z_j \tag{6.4}$$

can be equivalently described as

$$S = \sum_{i,j} E\left[\frac{z_i \cdot \Delta x_i}{\sigma_i} \cdot \frac{z_j \cdot \Delta x_j}{\sigma_j}\right] \tag{6.5}$$

i.e., $S = E[s]$, where

$$s \stackrel{\text{def}}{=} \sum_{i,j} \frac{z_i \cdot \Delta x_i}{\sigma_i} \cdot \frac{z_j \cdot \Delta x_j}{\sigma_j} \tag{6.6}$$

The expression s is hence the square

$$s = \left(\sum_i \frac{z_i \cdot \Delta x_i}{\sigma_i} \right)^2 \tag{6.7}$$

This square makes s always non-negative, thus its expected value $S = E[s]$ is also always non-negative, so the correlation matrix is positive semi-definite.

To quantify the uncertainty of the expert estimated correlation and obtain a positive semi-definite correlation matrix, three scenarios are considered:

Scenario 1. When the elements in the correlation matrix are entirely provided by expert estimates $\widetilde{a}_{ij}$, the smallest possible value $\varepsilon > 0$ is desired for which there exists a positive semi-definite matrix a_{ij} satisfying:

$$|\widetilde{a}_{ij} - a_{ij}| \leq \varepsilon \tag{6.8}$$

where $a_{ij} = \widetilde{a}_{ij} = 1$ when $i = j$.

Scenario 2. It may also occur that a part of the correlation matrix uses the empirical correlations e_{ij}. Although any statistical estimates based on finite samples are also approximate, these statistical correlations are usually much more accurate than the expert estimates [131], and therefore used as known correlation coefficient. In this case, the smallest possible value $\varepsilon > 0$ is wanted for which there exists a positive semi-definite matrix satisfying:

- $a_{ij} = e_{ij}$ for all pairs $(i, j) \in S$;
- $|\widetilde{a}_{ij} - a_{ij}| \leq \varepsilon$ for all pairs $(i, j) \in X$.

where $a_{ij} = \widetilde{a}_{ij} = 1$ when $i = j$; S is the set of pairs using exact statistical correlation coefficients; X is the set of pairs using expert estimates.

Scenario 3. Experts can give not only numerical estimate but also a fuzzy estimate [49–51] describing their opinion about the correlation, where experts can mark the certainty of their estimate on a scale from 0 to 1 [132]. Let α be this degree of certainty and if there

are two degrees $\alpha < \alpha'$, the interval with the lower degree of certainty α comprises the interval with the higher degree of certainty α': $[a_{ij}^-(\alpha'), a_{ij}^+(\alpha')] \subseteq [a_{ij}^-(\alpha), a_{ij}^+(\alpha)]$. Thus, the largest α corresponds to the tightest interval $[a_{ij}^-(\alpha), a_{ij}^+(\alpha)]$ that includes the actual (unknown) correlation coefficient. For example, if an expert claims a correlation coefficient to be 0.7 and the expert's accuracy is ± 0.2, a fuzzy expression of this correlation coefficient can be $[0.5(0), 0.9(0)]$. As can be seen, this is a confident estimate of a positive correlation. On the other hand, if an expert claims a correlation coefficient to be 0.2 and the expert's accuracy is ± 0.3, with fussy word $[-0.1(0), 0.5(0)]$, there may or may not be a positive correlation. Hence to establish a more certain correlation, the largest α is wanted to seek a tightest interval for which there exists a positive semi-definite matrix satisfying:

- $a_{ij} = e_{ij}$ for all pairs $(i, j) \in S$;
- $a_{ij} \in [a_{ij}^-(\alpha), a_{ij}^+(\alpha)]$ for all pairs $(i, j) \in X$.

where $a_{ij} = \widetilde{a}_{ij} = 1$ when $i = j$; S is the set of pairs using exact statistical correlation coefficients; X is the set of pairs using expert estimates.

6.3 Reformulation in terms of eigenvalues

The aim now is to find a positive semi-definite matrix with elements a_{ij}, existing in the ε-vicinity of the given invalid correlation matrix. The definition of a positive semi-definite matrix requires $\mathbf{z}^T\mathbf{A}\mathbf{z} \geq 0$ for any possible vector $\mathbf{z}$. A sufficient and necessary condition of the positive semi-definite matrix is that its all eigenvalues are non-negative.

Since expert estimates of the correlation coefficients are approximate, they may form an invalid correlation matrix, and some of their eigenvalues will be negative. If instead of looking for the largest absolute difference $|\widetilde{a}_{ij} - a_{ij}|$, looking for the mean squared difference solves the problem by correcting the negative eigenvalues and remaining diagonal elements to 1s [131, 133–138]. However, a small mean squared difference cannot prevent a large individual change $|\widetilde{a}_{ij} - a_{ij}|$ in some particular pairs (i, j). The individual smallest negative eigenvalue is the most important, which differs the most the

negative definite correlation matrix from the valid positive semi-definite one. Therefore the invalid correlation matrix is modified via making the smallest negative eigenvalue be 0.

6.3.1 Correlation matrix sensitivity to eigenvalues

Scenario 1. is considered first, where all elements of the correlation matrix are provided by expert estimates and the correlation matrix is negative definite. Let $a_{ij}^{(0)}$ be the elements of an invalid correlation matrix, $\lambda < 0$ $(\lambda \in \lambda_i, i = 1, ..., n)$ be the smallest negative eigenvalue, and $\mathbf{e_i}$ be the corresponding unit eigenvector. The unit vector $\mathbf{e_i}$ is such that:

$$\sum_{j=1}^{n} e_j^2 = 1 \tag{6.9}$$

This is an eigenvector corresponding to the eigenvalue λ, which, by definition:

$$\mathbf{A}\mathbf{e}_i = \lambda \mathbf{e}_i \tag{6.10}$$

this in turns means:

$$\sum_j a_{ij}^{(0)} \cdot e_j = \lambda \cdot \mathbf{e}_i \tag{6.11}$$

Expert estimates are approximate but are assumed with small errors because they are trusted experts. Hence the inaccuracy is relatively small, quadratic terms of this inaccuracy can be safely ignored. The elements of the updated matrix $a_{ij}^{(0)} + \Delta a_{ij}$ should be close to the elements of the original matrix $a_{ij}^{(0)}$, so that the corresponding unit eigenvector should be close to the original eigenvector e_i. Thus, the corresponding elements in unit eigenvector of the updated matrix can be written as $e_j + \Delta e_j$, where the deviations Δe_j are assumed to be small.

The new vector $\mathbf{e}_i + \Delta \mathbf{e}_i$ is still a unit vector, which means:

$$\sum_{j=1}^{n} (e_j + \Delta e_j)^2 = \sum_{j=1}^{n} \left(e_j^2 + 2e_j \cdot \Delta e_j + (\Delta e_j)^2 \right) = 1 \tag{6.12}$$

Open the parenthesis:

$$\sum_{j=1}^{n} e_j^2 + 2\sum_{j=1}^{n} e_j \cdot \Delta e_j + \sum_{j=1}^{n} (\Delta e_j)^2 = 1 \tag{6.13}$$

Equation 6.13 can be simplified via:

- quadratic terms $(\Delta e_j)^2$ can be safely ignored due to the assumption of small inaccuracy for trusted expert estimates;
- the first term is equal to 1 since $\sum_{j=1}^{n} e_j^2 = 1$.

Thus, eventually Equation 6.13 gives:

$$\sum_{j=1}^{n} e_j \cdot \Delta e_j = 0 \tag{6.14}$$

In a geometric prospective, Equation 6.14 means that the deviation vector Δe_i is orthogonal to the original eigenvector e_i. This is a requirement that would be used in later calculations.

Using the eigenvalue's definition (Equation 6.11), the fact that modified eigenvalue is 0 means:

$$\sum_{j=1}^{n} (a_{ij}^{(0)} + \Delta a_{ij}) \cdot (e_j + \Delta e_j) = 0 \tag{6.15}$$

Open the parentheses:

$$\sum_{j=1}^{n} a_{ij}^{(0)} \cdot e_j + \sum_{j=1}^{n} \Delta a_{ij} \cdot e_j + \sum_{j=1}^{n} a_{ij}^{(0)} \cdot \Delta e_j + \sum_{j=1}^{n} \Delta a_{ij} \cdot \Delta e_j = 0 \tag{6.16}$$

Equation 6.16 can be simplified via:

- quadratic terms $\Delta a_{ij} \cdot \Delta e_j$ can be safely ignored due to the assumption of small inaccuracy for trusted expert estimates;
- the first term is equal to $\lambda \cdot \mathbf{e}_i$ since Equation 6.11.

Thus, eventually Equation 6.15 can be written as:

$$\sum_{j=1}^{n} a_{ij}^{(0)} \cdot \Delta e_j = |\lambda| \cdot e_i - \sum_{j=1}^{n} \Delta a_{ij} \cdot e_j \tag{6.17}$$

A solution Δe_j of this system of linear equations orthogonal to $\mathbf{e}_i$ is now desired. Let $\mathbf{f}_i$ be:

$$\mathbf{f}_i \stackrel{\text{def}}{=} |\lambda| \cdot \mathbf{e_i} - \sum_{j=1}^{n} \Delta a_{ij} \cdot e_j \tag{6.18}$$

which also has the form:

$$\sum_{j=1}^{n} a_{ij}^{(0)} \cdot \Delta e_j = \mathbf{f_i} \tag{6.19}$$

In order to find such a solution, a new orthonormal basis is used. The components of the vectors $\mathbf{f}_i$ and $\Delta\mathbf{e}_i$ are denoted in the new basis by $\mathbf{F_k}$ and $\mathbf{\Delta E_k}$ and the components of the matrix $a_{ij}^{(0)}$ in the new basis are denoted by $A_{k\ell}^{(0)}$. $A_{k\ell}^{(0)}$ can be denoted by the old orthonormal basis, taking a diagonal form:

$$A_{k\ell}^{(0)} = \lambda_k \cdot \delta_{k\ell} \tag{6.20}$$

where λ_k is the k-th eigenvalue and $\delta_{k\ell}$ is the Kronecker symbol, i.e.:

- $\delta_{kk} = 1$ for all k;
- $\delta_{k\ell} = 0$ for all $k \neq \ell$.

Thus, in the new orthonormal basis, following Equation 6.19, the linear equations can be represented by:

$$\lambda_k \cdot \Delta E_k = F_k \tag{6.21}$$

The solution to this new system of equation is straightforward:

$$\Delta E_k = \frac{F_k}{\lambda_k} \tag{6.22}$$

Thus, in the original basis, the solution has the form:

$$\Delta e_j = \sum_{k=1}^{n} \frac{F_k}{\lambda_k} \cdot e_j^{(k)} \tag{6.23}$$

where $e_j^{(k)}$ is the eigenvector corresponding to the k-th eigenvalue. In other words, the solution is a linear combination of different eigenvectors.

Remembering the requirement that this solution should be orthogonal to the eigenvector (Equation 6.14), thus the component in F_k should be 0, namely, $\sum_1^n \mathbf{f_i e_i} = 0$. Considering Equation 6.19, hence:

$$0 = \sum_{i=1}^{n} |\lambda| \cdot \mathbf{e_i}^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} \Delta a_{ij} \cdot e_i \cdot e_j \tag{6.24}$$

Also for the unit eigenvector $\mathbf{e_i}$:

$$\sum_{j=1}^{n} e_j^2 = 1 \tag{6.25}$$

Thus Equation 6.24 yields to:

$$\sum_{i \neq j} \Delta a_{ij} \cdot e_i \cdot e_j = |\lambda| \tag{6.26}$$

This is how the correlation matrix changes if changing the smallest eigenvalue λ to 0. But to express the modified correlation matrix, how each element changes individually needs to be calculated.

6.3.2 The change of individual elements

Recapping that the smallest possible value $\varepsilon > 0$ is desired, for which there exists conditions $|\Delta a_{ij}| \leq \varepsilon$ and $\sum_{i \neq j} \Delta a_{ij} \cdot e_i \cdot e_j = |\lambda|$.

Due to $|\Delta a_{ij}| \leq \varepsilon$, it follows that:

$$|\Delta a_{ij} \cdot e_i \cdot e_j| \leq \varepsilon \cdot |e_i| \cdot |e_j| \tag{6.27}$$

This can be connected to Equation 6.26:

$$|\lambda| = \left| \sum_{i \neq j} \Delta a_{ij} \cdot e_i \cdot e_j \right| \leq \sum_{i \neq j} |\Delta a_{ij} \cdot e_i \cdot e_j| \leq \sum_{i \neq j} \varepsilon \cdot |e_i| \cdot |e_j| \tag{6.28}$$

Let S_0 denote:

$$S_0 \stackrel{\text{def}}{=} \sum_{i \neq j} |e_i| \cdot |e_j| \tag{6.29}$$

Putting the expression of Equation 6.28 in another way: if $\varepsilon \cdot S_0 < |\lambda|$, i.e., if

$$\varepsilon < \frac{|\lambda|}{S_0} \tag{6.30}$$

Then Equation 6.26 cannot be satisfied. Thus, it is sufficient to take the change of the individual element Δa_{ij} as follows:

$$\Delta a_{ij} = \varepsilon \cdot sign(e_i) \cdot sign(e_j) \tag{6.31}$$

Namely, the change Δa_{ij} is maximum when:

$$\sum_{i \neq j} \Delta a_{ij} \cdot e_i \cdot e_j = \varepsilon \cdot \sum_{i \neq j} \text{sign}(e_i) \cdot \text{sign}(e_j) \cdot e_i \cdot e_j \tag{6.32}$$

The corresponding smallest possible ε is as follows:

$$\varepsilon = \frac{|\lambda|}{S_0} \tag{6.33}$$

where $\text{sign}(x)$ means the sign of the value x:

- $\text{sign}(x) = 1$ when $x > 0$:
- $\text{sign}(x) = -1$ when $x < 0$.

In a word, for every number $x \in \mathbb{R}$: $x \cdot \text{sign}(x) = |x|$.

Furthermore, the computation of S_0 can be simplified as follows:

$$S_0 = \sum_{i \neq j} |e_i| \cdot |e_j| = \sum_{i,j} |e_i| \cdot |e_j| - \sum_i |e_i|^2 \tag{6.34}$$

The first sum in the right-hand side equals to the product

$$\left(\sum_{i=1}^{n} |e_i| \right) \cdot \left(\sum_{j=1}^{n} |e_j| \right) \tag{6.35}$$

which equals to the square

$$\left(\sum_{i=1}^{n} |e_i| \right)^2 \tag{6.36}$$

In the end, S_0 is simplified as:

$$S_0 = \left(\sum_{i=1}^{n} |e_i| \right)^2 - \sum_{i=1}^{n} e_i^2 \tag{6.37}$$

6.3.3 Considering available empirical correlations

In section 6.2, Scenario 2. referred to that there exists a set S, where for pairs $(i, j) \in S$, empirical correlations denoted by e_{ij} are available. The elements of the given invalid correlation matrix has the form:

- $a_{ij} = 1$ when $i = j$;
- $a_{ij} = e_{ij}$ when $(i, j) \in S$;
- $a_{ij} = \widetilde{a}_{ij}$ when $(i, j) \notin S$.

For the pairs (i, j) do not belong to set S, the above solution for Scenario 1. is still applicable. For some pairs (i, j) belong to set S, both empirical correlations e_{ij} and expert estimates $\widetilde{a}_{ij}$ are available, the empirical correlations e_{ij} are safely assumed as known actual values. Hence the difference between empirical correlations and expert estimates should be taken into account. The smallest ε takes the form:

$$\max \left(\frac{|\lambda|}{S_0}, \max_{(i,j) \in S \cap X} |\widetilde{a}_{ij} - e_{ij}| \right) \tag{6.38}$$

6.4 Resulting algorithm for Scenario 1. and 2.

The resulting algorithm for Scenario1. and 2. can be summarized as follows. At first, a correlation matrix is given:

- for some pairs (i, j), the expert estimates $\widetilde{a}_{ij}$ of the covariances are given. The set of all such pairs will be denoted by X. X may cover the whole correlation matrix.
- for some pairs (i, j), the values e_{ij} of the empirical covariances are given. The set of all such pairs is denoted by S. S may not exist: $S = \emptyset$. Elements in the set S may or may not in the set X:
  - when $S \cap X = \emptyset$, i.e., for every pair, either an empirical covariance or an expert estimate is provided, but not both;
  - when $S \cap X \neq \emptyset$, i.e., for some pairs, both the empirical value of the covariance and the expert estimate are provided.

Summarizing, a correlation matrix is known by either an empirical value or an expert estimate is given to the element of the matrix, or both.

The aim is to quantify the accuracy of the expert estimates, i.e., to find the smallest $\varepsilon > 0$ for which $\widetilde{a}_{ij}$ in the ε vicinity, there exists a non-negative definite correlation matrix.

The solution is provided by the correlation matrix's eigenvalues and eigenvectors:

Algorithm for Scenario 1. & 2.

1. Represent the matrix with elements $a_{ij}^{(0)}$ as follows:

- for $i = j$, $a_{ij}^{(0)} = 1$;
- for $(i, j) \in S$, $a_{ij}^{(0)} = e_{ij}$;
- for $(i, j) \notin S$ and $i \neq j$, $a_{ij}^{(0)} = \widetilde{a}_{ij}$.

2. Compute the smallest eigenvalue λ. The following actions depend on whether this smallest eigenvalue is non-negative or negative.

2.1. If $\lambda \geq 0$, the matrix $a_{ij}^{(0)}$ is already non-negative definite.

2.1.1. When $S \cap X = \emptyset$, this means no conclusions can be made about the accuracy of the expert estimates: it could be that the expert estimates are exact.

2.1.2. When $S \cap X \neq \emptyset$, the largest difference between the expert estimates and the empirical correlations is considered as an estimate for expert accuracy:

$$\varepsilon = \max_{(i,j) \in S \cap X} |\widetilde{a}_{ij} - e_{ij}|.$$

2.2. If $\lambda < 0$, compute the corresponding unit eigenvector to express the wanted ε:

$$S_0 = \sum_{(i,j) \notin S \,\&\, i \neq j} |e_i| \cdot |e_j| \tag{6.39}$$

Especially, when $S = \emptyset$, computing S_0 can be simplified by:

$$S_0 = \left(\sum_{i=1}^{n} |e_i| \right)^2 - \sum_{i=1}^{n} e_i^2 \tag{6.40}$$

The resulting estimate for ε depends on the case:

2.2.1. When $S \cap X = \emptyset$,

$$\varepsilon = \frac{|\lambda|}{S_0} \tag{6.41}$$

2.2.2. When $S \cap X \neq \emptyset$,

$$\varepsilon = \max \left(\frac{|\lambda|}{S_0}, \max_{(i,j) \in S \cap X} |\widetilde{a}_{ij} - e_{ij}| \right) \tag{6.42}$$

6.5 Fuzzy estimates

Experts often describe their uncertainty in terms of fuzzy estimates. Fuzzy estimates $[a_{ij}^-(\alpha), a_{ij}^+(\alpha)]$ give an upper and lower limit of the estimated correlation coefficient that contain expert's certainty levels. To know how much the estimated intervals deviate from a valid correlation matrix, fuzzy intervals with the following conditions are desired:

Condition 1. there exist values a_{ij} within the intervals, which can form a positive semi-definite correlation matrix.

Condition 2. for some pairs, where both empirical correlations and expert estimates are available, the empirical correlations lie within the corresponding intervals, that is for all such pairs:

$$a_{ij}^-(\alpha) \leq e_{ij} \leq a_{ij}^+(\alpha) \tag{6.43}$$

Since the intervals $[a_{ij}^-(\alpha), a_{ij}^+(\alpha)]$ grow when α decreases, if the above two conditions are satisfied for some α, they are also satisfied for all smaller values $\alpha' < \alpha$ as well. Therefore to find the smallest change to the given correlation intervals, the aim now is to find the largest α for which both above conditions are satisfied. A bisection idea is applied to solve this problem:

- First, check whether both conditions are satisfied for $\alpha = 1$. If they are satisfied, then $\alpha = 1$ is the wanted value.
- If one or both of the above conditions are not satisfied for $\alpha = 1$, then check whether they are satisfied for $\alpha = 0$:
  - If they are not even satisfied for $\alpha = 0$, this means that the expert underestimates his/her uncertainty, so this fuzzy information is not reliable.
  - If both conditions are satisfied for $\alpha = 0$, there then exists:
    - a value $\underline{\alpha}$ for which both conditions are satisfied;
    - a value $\overline{\alpha}$ for which at least one of the conditions is not satisfies.

In this case, the desired value α is somewhere between $\underline{\alpha}$ and $\overline{\alpha}$, i.e., somewhere on the interval

$$[\underline{\alpha}, \overline{\alpha}]$$

Then check whether both conditions are satisfied for the midpoint of this interval:

$$\alpha_m = \frac{\underline{\alpha} + \overline{\alpha}}{2} \tag{6.44}$$

To get a positive semi-definite correlation matrix, some correlation coefficients need to be adjusted to larger values and others to smaller values. Both sides need to have enough room to afford the change. This is the reason why the midpoint is chosen.

- If both conditions are satisfied, a new interval $[\alpha_m, \overline{\alpha}]$ of half size of $[\underline{\alpha}, \overline{\alpha}]$ is obtained, which contains the desired value α. Then replace $\underline{\alpha}$ with α_m while keeping $\overline{\alpha}$ unchanged.
- On the other hand, if at least one of the conditions is not satisfied, a new interval $[\underline{\alpha}, \alpha_m]$ of half size of $[\underline{\alpha}, \overline{\alpha}]$ is also obtained, which contains the desired α. Then replace $\overline{\alpha}$ with α_m while keeping $\underline{\alpha}$ unchanged.

In both cases, the size of the interval is decreased in half.

The above bisection process is iterated until a precision limit is reached. It is assumed that practically, experts do not describe their degree of certainty in a higher accuracy more than one decimal digit. Starting with an interval $[0, 1]$ of width 1, an interval of width $1/2^4 = 1/16 = 0.0625$ is obtained after 4 iterations. Hence, 4 iterations are more than enough for finding the main digit of the desired value α.

The remaining question is that how to check whether the aforementioned both conditions are satisfied within $[a_{ij}^-(\alpha), a_{ij}^+(\alpha)]$ for a given α. Checking Condition 2. is checking the corresponding inequalities directly. Checking Condition 1. needs to check the existence of a positive semi-definite matrix. It still makes sense to test a representative numerical value—the midpoint of the interval:

$$\widetilde{a}_{ij} = \frac{a_{ij}^-(\alpha) + a_{ij}^+(\alpha)}{2} \tag{6.45}$$

Modifying a negative definite matrix $\widetilde{a}_{ij}$ to be a positive semi-definite one, as has been shown, means that $\sum_{i \neq j} \Delta a_{ij} \cdot e_i \cdot e_j = |\lambda|$ (Equation 6.26) has to be true. Since at least such an Δa_{ij} should be found within the interval of a given α, the aim is to know the maximum change Δa_{ij} can take within the interval. Let v_{ij} denote the largest possible Δa_{ij},

- When the product $e_i \cdot e_j$ is positive, the maximum of this term is attained for positive values Δa_{ij}. The largest positive value v_{ij} of Δa_{ij} is equal to $a_{ij}^{+}(\alpha) - \widetilde{a}_{ij}$.
- When the product $e_i \cdot e_j$ is negative, the maximum of the i-th term in the sum is attained for negative values Δa_{ij}. The largest absolute value of these negative values is $v_{ij} = \widetilde{a}_{ij} - a_{ij}^{-}(\alpha)$.

If $\sum_{i \neq j} v_{ij} \cdot |e_i| \cdot |e_j| \geq |\lambda|$, the existence of a positive semi-definite matrix is proven.

6.6 Fuzzy algorithm

The resulting algorithm for the fuzzy scenario can be summed up as follows. A correlation matrix with fuzzy estimates is given:

- For some pairs (i, j), the empirical correlations e_{ij} are given. The set of all such pairs is denoted by S;
- For some pairs (i, j), experts give fuzzy estimates $[a_{ij}^{-}(\alpha), a_{ij}^{+}(\alpha)]$ corresponding to different values α. The set of all such pairs (i, j) is denoted by X.

Summarizing, for every pair of different indices, either an empirical value or an expert estimate is know, or both.

The aim is to return the largest possible value α so that, for each $(i, j) \notin S$ for which $i \neq j$, the interval $[a_{ij}^{-}(\alpha), a_{ij}^{+}(\alpha)]$ is used as the range of possible values of valid correlation. In addition, for some elements in the correlation matrix with known empirical correlations, the empirical correlations should lie within the corresponding expert-estimated fuzzy

intervals $a_{ij}^-(\alpha) \le e_{ij} \le a_{ij}^+(\alpha)$. These can be considered as two conditions that the desired fuzzy interval should meet.

The algorithm is as follows:

Algorithm for Scenario 3.

1. For each $(i,j) \notin S$ for which $i \ne j$, compute the value

$$\widetilde{a}_{ij} = \frac{a_{ij}^-(1) + a_{ij}^+(1)}{2} \tag{6.46}$$

2. Represented the matrix with elements $a_{ij}^{(0)}$ as follows:

- for $i = j$, $a_{ij}^{(0)} = 1$;
- for $(i,j) \in S$, $a_{ij}^{(0)} = e_{ij}$;
- for $(i,j) \notin S$ and $i \ne j$, $a_{ij}^{(0)} = \widetilde{a}_{ij}$.

3. Compute the smallest eigenvalue λ of the matrix $a_{ij}^{(0)}$ and the corresponding unit eigenvector e_i.

3.1. Check whether for $\alpha = 1$:

3.1.1. For all $(i,j) \in S \cap X$:

$$a_{ij}^-(\alpha) \le e_{ij} \le a_{ij}^+(\alpha) \tag{6.47}$$

3.1.2. If at least one of these inequalities is not satisfied, conclude that the conditions are not satisfied.

3.1.3.1. If all the above inequalities are satisfied and $\lambda \ge 0$, conclude that both conditions are satisfied.

3.1.3.2. If all the above inequalities are satisfied but $\lambda < 0$, for all $(i,j) \notin S$ for which $i \ne j$, compute the following value v_{ij}:

- when $\text{sign}(e_i) \cdot \text{sign}(e_j) > 0$,

$$v_{ij} = a_{ij}^{+}(\alpha) - \widetilde{a}_{ij} \tag{6.48}$$

- when $\text{sign}(e_i) \cdot \text{sign}(e_j) < 0$

$$v_{ij} = \widetilde{a}_{ij} - a_{ij}^{-}(\alpha) \tag{6.49}$$

Check whether:

$$\sum_{(i,j)\notin S \,\&\, i\neq j} v_{ij} \cdot |e_i| \cdot |e_j| \geq |\lambda| \tag{6.50}$$

- If this inequality is satisfied, both conditions are satisfied;
- If this inequality is not satisfied, conclude that this interval does not satisfy all conditions.

3.2. If the conditions are concluded as satisfied for $\alpha = 1$, the desired interval is $[a_{ij}^{-}(1), a_{ij}^{+}(1)]$. If not, check whether they are satisfied for $\alpha = 0$. If they are not even satisfied for $\alpha = 0$, all the fuzzy information is ignored. In this case, only the representative value $\widetilde{a}_{ij}$ is kept as the expert estimate and used as the non-fuzzy scenarios that has been discussed in section 6.4.

3.3. If the conditions are satisfied for $\alpha = 0$ and not satisfied for $\alpha = 1$, let $\underline{\alpha} = 0$, $\overline{\alpha} = 1$, $\alpha_m = \frac{\underline{\alpha}+\overline{\alpha}}{2}$ and start iterations.

On each iteration, check whether the conditions are satisfied for α_m:

- If the conditions are satisfied for α_m, replace $\underline{\alpha}$ with α_m, while keeping $\overline{\alpha}$ unchanged.
- If the conditions are not satisfied for α_m, replace $\overline{\alpha}$ with α_m while keeping $\underline{\alpha}$ unchanged.

Let δ denote the precision requirement. Iterations stop when $\overline{\alpha} - \underline{\alpha} \leq \delta$ for a given δ (e.g., for $\delta = 0.1$). At this point, the midpoint is the desired value α:

$$\alpha_m = \frac{\underline{\alpha} + \overline{\alpha}}{2} \tag{6.51}$$

6.7 Numerical example based on climb-cruise engine matching

Continue on the climb-cruise engine marching problem, the correlations between the resulting model inputs and observations are of interest. Studying correlations help to understand the physical problem, such as whether causal or not between a certain input and a certain observation. There are 9 model inputs and 4 observations in this problem. The correlation matrix between model inputs and observations is denoted by $\mathbf{A}^{(0)}$ of size 13×13, which is composed by the correlation coefficients $a_{ij}^{(0)}$ within model inputs when $i, j = 1, ..., 9$, between model inputs and observations when $i = 1, .., 9$ & $j = 10, ..., 13$ or $i = 10, .., 13$ & $j = 1, ..., 9$, and within observations when $i, j = 10, ..., 13$. Hence, the correlation matrix is correspondingly decomposed into four pieces, denoted by:

$$\begin{bmatrix} \mathbf{A}_1^{(0)}(a_{1_{ij}}^{(0)}, i, j = 1, ..., 9) & \mathbf{A}_2^{(0)}(a_{2_{ij}}^{(0)}, i = 1, .., 9 \& j = 10, ..., 13) \\ \mathbf{A}_3^{(0)}(a_{3_{ij}}^{(0)}, i = 10, .., 13 \& j = 1, ..., 9) & \mathbf{A}_4^{(0)}(a_{4_{ij}}^{(0)}, i, j = 10, ..., 13) \end{bmatrix}$$

The correlation coefficients within model inputs can be computed directly by the resulting input samples:

$$\mathbf{A}_1^{(0)} = \begin{bmatrix} 1 & -0.1791 & -0.0767 & -0.0629 & -0.4908 & 0.3015 & 0.1632 & 0.1887 & 0.0308 \\ -0.1791 & 1 & -0.0923 & -0.0981 & -0.0028 & -0.2031 & 0.2559 & 0.3903 & 0.0913 \\ -0.0767 & -0.0923 & 1 & 0.1466 & 0.5497 & 0.2956 & 0.0679 & -0.1571 & 0.187 \\ -0.0629 & -0.0981 & 0.1466 & 1 & 0.0732 & 0.1133 & 0.1304 & 0.2218 & 0.0667 \\ -0.4908 & -0.0028 & 0.5497 & 0.0732 & 1 & -0.4354 & -0.0505 & -0.6239 & -0.0112 \\ 0.3015 & -0.2031 & 0.2956 & 0.1133 & -0.4354 & 1 & 0.3858 & 0.3322 & 0.1884 \\ 0.1632 & 0.2559 & 0.0679 & 0.1304 & -0.0505 & 0.3858 & 1 & -0.1551 & -0.1284 \\ 0.1887 & 0.3903 & -0.1571 & 0.2218 & -0.6239 & 0.3322 & -0.1551 & 1 & 0.0076 \\ 0.0308 & 0.0913 & 0.187 & 0.0667 & -0.0112 & 0.1884 & -0.1284 & 0.0076 & 1 \end{bmatrix}$$

However due to the limited observations (only one for each type of them), the correlation coefficients between the resulting inputs and observations or within the observations can not be computed. In this case, they can be filled in with expert estimates, assuming the values are as follows:

$$\mathbf{A}_2^{(0)} = \begin{bmatrix} -0.2442 & -0.1299 & -0.1053 & 0 \\ -0.1877 & -0.1586 & 0.0820 & -0.0147 \\ -0.1309 & 0.0767 & 0.0672 & -0.0288 \\ -0.0121 & -0.1332 & 0.1464 & 0.1332 \\ 0.0484 & -0.0407 & -0.0563 & -0.0286 \\ -0.1694 & -0.0885 & -0.0289 & -0.0834 \\ -0.0960 & -0.2609 & 0.0776 & -0.2000 \\ -0.2642 & -0.0535 & -0.0111 & -0.1028 \\ -0.1280 & -0.0666 & 0.0205 & 0.1000 \end{bmatrix}$$

$$\mathbf{A}_4^{(0)} = \begin{bmatrix} 1 & -0.0901 & -0.1541 & -0.0171 \\ -0.0901 & 1 & 0.0522 & -0.2134 \\ -0.1541 & 0.0522 & 1 & -0.1000 \\ -0.0171 & -0.2134 & -0.1000 & 1 \end{bmatrix}$$

The correlation matrix is symmetric, hence matrix $\mathbf{A}_3^{(0)}$ is the transpose of matrix $\mathbf{A}_2^{(0)}$. The eigenvalues for the whole correlation matrix $\lambda_i (i = 1, ..., 13)$ are:

$$-0.0175, 0.0422, 0.4468, 0.5943, 0.6483, 0.8642, 0.9549, 1.1103, 1.3439, 1.3616, 1.4771, 1.7683, 2.4257$$

Because there is one negative eigenvalue $\lambda = -0.0175$, the correlation matrix is not non-negative definite. The corresponding eigenvector $\mathbf{e_i}$ is:

$$[-0.0660, 0.3331, 0.1843, 0.2272, -0.4880, 0.0987, -0.3522, -0.5775, -0.1440, -0.1200, -0.1239, -0.1085, -0.1774]^T$$

According to the algorithm for Scenario 1. and 2., when $i, j = 1, ..., 9$, $(i, j) \in S$, the empirical correlation coefficients $a^{(0)}_{1_{ij}}$ would not change. The other estimated correlation coefficients would be adjusted to obtain a positive semi-definite correlation matrix. The smallest possible change for elements $a^{(0)}_{ij}$, $(i, j) \notin S$ and $i \neq j$, is $\varepsilon = 0.0071$. Via the changes $\Delta a_{ij} = \varepsilon \cdot sign(e_i) \cdot sign(e_j)$, the mathematically valid positive semi-definite correlation matrix is then obtained:

$$\mathbf{A} = \begin{bmatrix}
1 & -0.1791 & -0.0767 & -0.0629 & -0.4908 & 0.3015 & 0.1632 \\
-0.1791 & 1 & -0.0923 & -0.0981 & -0.0028 & -0.2031 & 0.2559 \\
-0.0767 & -0.0923 & 1 & 0.1466 & 0.5497 & 0.2956 & 0.0679 \\
-0.0629 & -0.0981 & 0.1466 & 1 & 0.0732 & 0.1133 & 0.1304 \\
-0.4908 & -0.0028 & 0.5497 & 0.0732 & 1 & -0.4354 & -0.0505 \\
0.3015 & -0.2031 & 0.2956 & 0.1133 & -0.4354 & 1 & 0.3858 \\
0.1632 & 0.2559 & 0.0679 & 0.1304 & -0.0505 & 0.3858 & 1 \\
0.1887 & 0.3903 & -0.1571 & 0.2218 & -0.6239 & 0.3322 & -0.1551 \\
0.0308 & 0.0913 & 0.187 & 0.0667 & -0.0112 & 0.1884 & -0.1284 \\
-0.2371 & -0.1948 & -0.1380 & -0.0192 & 0.0555 & -0.1765 & -0.0889 \\
-0.1228 & -0.1657 & 0.0696 & -0.1403 & -0.0336 & -0.0956 & -0.2538 \\
-0.0982 & 0.0749 & 0.0601 & 0.1393 & -0.0492 & -0.0360 & -0.0847 \\
0.0071 & -0.0218 & -0.0359 & 0.1261 & -0.0215 & -0.0905 & -0.1929
\end{bmatrix}$$

$$\begin{bmatrix}
0.1887 & 0.0308 & -0.2371 & -0.1228 & -0.0982 & 0.0071 \\
0.3903 & 0.0913 & -0.1948 & -0.1657 & 0.0749 & -0.0218 \\
-0.1571 & 0.187 & -0.1380 & 0.0696 & 0.0601 & -0.0359 \\
0.2218 & 0.0667 & -0.0192 & -0.1403 & 0.1393 & 0.1261 \\
-0.6239 & -0.0112 & 0.0555 & -0.0336 & -0.0492 & -0.0215 \\
0.3322 & 0.1884 & -0.1765 & -0.0956 & -0.0360 & -0.0905 \\
-0.1551 & -0.1284 & -0.0889 & -0.2538 & -0.0847 & -0.1929 \\
1 & 0.0076 & -0.2571 & -0.0464 & -0.0040 & -0.0957 \\
0.0076 & 1 & -0.1209 & -0.0595 & 0.0276 & 0.1071 \\
-0.2571 & -0.1209 & 1 & -0.0830 & -0.1470 & -0.0100 \\
-0.0464 & -0.0595 & -0.0830 & 1 & 0.0593 & -0.2063 \\
-0.0040 & 0.0276 & -0.1470 & 0.0593 & 1 & -0.0929 \\
-0.0957 & 0.1071 & -0.0100 & -0.2063 & -0.0929 & 1
\end{bmatrix}$$

Now the eigenvalues are:

$$0, 0.0420, 0.4527, 0.5914, 0.6381, 0.8668, 0.9563, 1.1035, 1.3419, 1.3553, 1.4765, 1.7497, 2.4254$$

All eigenvalues are equal to or larger than 0, hence the correlation matrix is proven to be non-negative definite after the conversion, with the largest diversion of the expert's estimates $\varepsilon = 0.0071$.

Considering Scenario 3.—instead of numerical estimates, assuming fuzzy estimates are available, as follows:

$$\mathbf{A}_2^{(0)} = \begin{bmatrix} [-0.3442, -0.1442] & [-0.2299, -0.0299] & [-0.2053, -0.0053] & [-0.1000, 0.1000] \\ [-0.2877, -0.0877] & [-0.2586, -0.0586] & [-0.0180, 0.1820] & [-0.1147, 0.0853] \\ [-0.2309, -0.0309] & [-0.0233, 0.1767] & [-0.0328, 0.1672] & [-0.1288, 0.0712] \\ [-0.1121, 0.0879] & [-0.2332, -0.0332] & [0.0464, 0.2464] & [0.0332, 0.2332] \\ [-0.0516, 0.1484] & [-0.1407, 0.0593] & [-0.1563, 0.0437] & [-0.1286, 0.0714] \\ [-0.2694, -0.0694] & [-0.1885, 0.0115] & [-0.1289, 0.0711] & [-0.1834, 0.0166] \\ [-0.1960, 0.0040] & [-0.3609, -0.1609] & [-0.0224, 0.1776] & [-0.3000, -0.1000] \\ [-0.3642, -0.1642] & [-0.1535, 0.0465] & [-0.1111, 0.0889] & [-0.2028, -0.0028] \\ [-0.2280, -0.0280] & [-0.1666, 0.0334] & [-0.0795, 0.1205] & [0, 0.2000] \end{bmatrix}$$

$$\mathbf{A}_4^{(0)} = \begin{bmatrix} 1 & [-0.2901, -0.0901] & [-0.3541, -0.1541] & [-0.2171, -0.0171] \\ [-0.2901, -0.0901] & 1 & [-0.1478, 0.0522] & [-0.4134, -0.2134] \\ [-0.3541, -0.1541] & [-0.1478, 0.522] & 1 & [-0.3000, -0.1000] \\ [-0.0271, -0.0171] & [-0.4134, -0.2134] & [-0.3000, -0.1000] & 1 \end{bmatrix}$$

Every estimated interval for the correlation coefficient takes a width for 0.2. The uncertainty level α on the interval takes the following distribution (Figure 6.2):

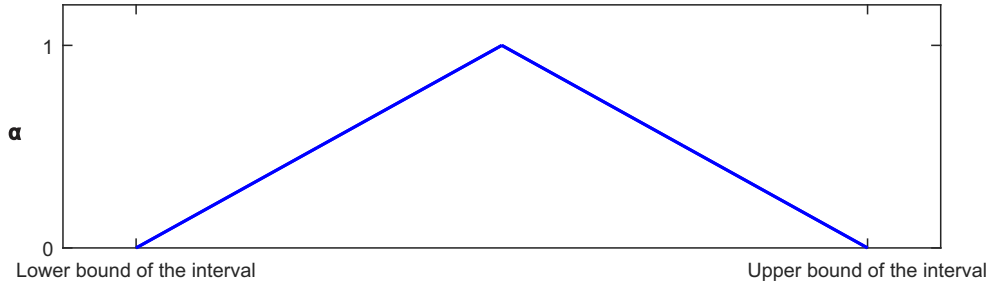


FIGURE 6.2: The distribution of the the uncertainty level α for the fuzzy estimate of each correlation coefficient.

Following the algorithm for Scenario 3., first check whether the two conditions are satisfied for $\alpha = 1$, namely the matrix $\mathbf{A}^{(0)}$ takes the midpoint in every interval for this example. Because the empirical correlations are considered unavailable due to limited observations, Condition 2. (Equation 6.43) is omitted. Condition 1. is checked by whether all eigenvalues for $\mathbf{A}^{(0)}(\alpha = 1)$ are non-negative. The eigenvalues for the whole

correlation matrix $\lambda_i, (i = 1, ..., n)$ are:

$$-0.0548, 0.0361, 0.3328, 0.5993, 0.6242, 0.8862, 0.9997, 1.0947, 1.3627, 1.4498, 1.4930, 1.7552, 2.4212$$

Because there is one negative eigenvalue $\lambda = -0.0548$, Condition 1. is not satisfied. Then check if both conditions are satisfied for $\alpha = 0$, namely the matrix $\mathbf{A}^{(0)}$ takes the bounds in every interval for this example. For the upper bound, the condition is satisfied, while for the lower bound the condition is not satisfied, which means there exists a positive semi-definite matrix in between that the condition can be satisfied. Hence the uncertainty of the fuzzy estimates can be quantified through finding the tightest intervals that containing a positive semi-definite correlation matrix. Following the proposed bisection algorithm for α, the size of the intervals is iteratively cut down to check whether a positive semi-definite matrix exists within $\mathbf{A}(\alpha)$. This is checked through Equation 6.50. For this example, during 4 iterations, α takes $0.5, 0.75, 0.875, 0.8125$ in turn. With a precision of to the fourth decimal digit, $\mathbf{A}(\alpha = 0.8125)$ is the desired matrix containing a positive semi-definite correlation matrix. $\mathbf{A}(\alpha = 0.8125)$ is too big to be included as a whole matrix in this page, thus it is shown by the components:

$$\mathbf{A}_2(\alpha = 0.8125) = \begin{bmatrix} [-0.3255, -0.1629] & [-0.2112, -0.0487] & [-0.1866, -0.0241] & [-0.0813, 0.0813] \\ [-0.2690, -0.1065] & [-0.2399, -0.0774] & [-0.0007, 0.1633] & [-0.096, 0.0666] \\ [-0.2122, -0.0497] & [-0.0046, 0.1579] & [-0.0141, 0.1484] & [-0.1101, 0.2144] \\ [-0.0934, 0.0692] & [-0.2145, -0.0520] & [0.0625, 0.2277] & [0.0519, 0.2144] \\ [-0.0329, 0.1297] & [-0.1220, 0.0405] & [-0.1376, 0.0250] & [-0.1098, 0.0527] \\ [-0.2507, -0.0882] & [-0.1698, -0.0073] & [-0.1102, 0.0523] & [-0.1647, -0.0022] \\ [-0.1773, -0.0148] & [-0.3422, -0.1796] & [-0.0037, 0.1589] & [-0.2813, -0.1188] \\ [-0.3455, -0.1830] & [-0.1348, 0.0277] & [-0.0924, 0.0702] & [-0.1841 - 0.0216] \\ [-0.2093, -0.0468] & [-0.1479, 0.0146] & [-0.0608, 0.1018] & [0.0188, 0.1813] \end{bmatrix}$$

$$\mathbf{A}_4(\alpha = 0.8125) = \begin{bmatrix} 1 & [-0.2714, -0.1089] & [-0.3354, -0.1728] & [-0.1984, -0.0359] \\ [-0.2714, -0.1089] & 1 & [-0.1291, 0.0335] & [-0.3947, -0.2322] \\ [-0.3354, -0.1728] & [-0.1291, 0.0335] & 1 & [-0.2813, -0.1188] \\ [-0.1984, -0.0359] & [-0.3947, -0.2322] & [-0.2813, -0.1188] & 1 \end{bmatrix}$$

$\mathbf{A}_1(\alpha = 0.8125)$ is the empirical correlation matrix for model inputs, which remain

unchanged. $\mathbf{A}_3(\alpha = 0.8125)$ is the transpose of matrix $\mathbf{A}_2(\alpha = 0.8125)$. Note that the assumed expert estimates in this example is not real, but they illustrate the implementation of the proposed algorithms. If real expert estimates are provided, of which the uncertainty would be quantified using the proposed algorithms, resulting in a valid, positive semi-definite and interrelated correlation matrix.

6.8 Conclusions

In this chapter, the uncertainty of the expert-estimated correlation is quantified, in a way of satisfying the mathematical meaning of a valid correlation matrix—being non-negative definite. Algorithms are provided to give a valid correlation matrix out of an invalid one. An invalid correlation matrix can be an expert-estimated correlation matrix describing the dependency between the given observation and its fitting inputs produced by HM. It also can be any negative definite correlation matrix relying on expert estimates, due to not enough data.

Multiple scenarios are considered:

Scenarios	Characteristics
Scenario 1.	Elements in the correlation matrix are entirely provided by expert estimates.
Scenario 2.	Empirical correlations are available for some elements in the correlation matrix.
Scenario 3.	Fuzzy estimates are provided instead of numerical estimates

TABLE 6.1: The considered scenarios for producing a non-negative definite correlation matrix.

Based on the SSHM result of the 'climb-cruise engine matching' problem, the uncertainty quantification of the expert-estimated correlation matrix was illustrated. Summarizing, the resulting algorithms provide novel solutions when dealing with an invalid correlation matrix. They present a valid non-negative definite correlation matrix which deviates the least from the given invalid correlation matrix.

Chapter 7

Conclusion

7.1 Summary of the work

The prediction ability of complex computer models (simulators) relies critically on how well they are calibrated to experimental data. That is to find the fitting input domain, which can provide a match between model outputs and given observations. The traditional calibration method is testing the goodness-of-fit between model outputs and given observations for each input sample, which is time-consuming for high dimensional and complex computer models. Bayesian model calibration has been popularized for solving this problem. But the disadvantage is that Bayesian model calibration makes probabilistic statements about the most likely inputs hence always produces a 'fitting' input domain even with a wrong model. History matching improves in this respect. It can diagnose that a model is not adequate by determining the whole input domain as implausible and as a result producing an empty input domain. Therefore this thesis focuses on this specific form of model calibration—history matching (HM).

Given observations, HM can fast eliminate vast implausible volumes of the input space by only focusing on the non-implausible input domain. However, when there is only a few observations or even only one observation, the non-implausible input space can be orders of magnitude smaller than the original input space. It is challenging when sampling on such a non-implausible domain, in addition this domain can be disconnected

or with a complex topology. The proposed SSHM provides a solution by using an engineering reliability method—Subset Simulation, which can screen the input domain by decomposing a rare non-implausible region into sequential less rare nested regions that are easier to sample at. Different MCMC sampling schemes within Subset Simulation were compared in the framework of SSHM. The adaptive MCMC with optimal scaling performs better because it generally produces more distinct non-implausible samples.

In order to test the proposed method, a real-world industrial problem was solved by the proposed SSHM. The 'climb-cruise engine matching' is originally a robust design problem, which tries to find appropriate airframe and engine parameters aiming to make less noise, less emission and use less fuel. If the targets with uncertainty margins were treated as observations with measurement uncertainty, the robust design problem can be viewed as a model calibration problem for multi-dimensional and complex simulators. In this context, it was demonstrated that SSHM can be applied to solve this problem. Based on simulators provided by the AirCADia framework, SSHM successfully found the fitting parameters for the wing aspect ratio, the sea level static thrust, the fan pressure ratio, the overall pressure ratio, the bypass ratio, the fuel flow factor and the overall drag, to provide a match between model outputs and the given targets/observations for the flyover noise, the sideline noise, the Nitrogen Oxide emissions and the block fuel.

Considering further improvements for SSHM, new training data in each wave were proposed to be selected from non-implausible samples in its last wave. This approach avoided screening the input space to select training samples on the non-implausible domain. Since the non-implausible input domain may appear disconnected and each region needs new training data to update the emulator, a classification method—DBSCAN was adopted to identify each region. New training data were then generated on each cluster, with the number of them proportional to the size of the cluster. In this way, the new training data can evenly distribute on each region of the disconnected non-implausible domain, which helped to improve the accuracy of the history matching for every non-implausible region.

When exploring the dependency between the resulting inputs of HM and observations, the correlation matrix may not be statistically calculated because of not enough data

for observations. Considering to rely on expert estimates, a method to quantify the uncertainty of expert-estimated correlation matrix was proposed. This method can give a mathematically valid correlation matrix, which deviates the least from the invalid (negative definite) expert-estimated correlation matrix. Fuzzy situations were considered when experts giving an interval estimate comprising his/her confidence levels. In that case, the tightest interval comprising a valid correlation matrix can be obtained through the proposed algorithm.

In conclusion, SSHM efficiently identifies the fitting input domain that can provide a match between model outputs and observations. SSHM helps to identify different sources of uncertainty existing along the model calibrating process. It also helps to improve the understanding of the modeled physical process by exploring the correlation between model inputs and observations. Except from model calibration, SSHM has been proven useful in robust design by treating the target as observation. With the connection of Subset Simulation, HM may be applied into reliability analysis to find the input domain that gives a large failure probability for complex engineering systems. Hopefully in the future, SSHM can be more widely used in engineering practice.

7.2 Discussion

This work has used Latin hypercube sampling when generating samples for the initial wave. It is also possible to use Quasi-Monte Carlo methods [139] such as Sobol's sequences [140] or Halton's sequences [141] to perform the intial design for simulation, as long as the ensemble of samples is representative of the input space. The Monte Carlo sampling (traditional random sampling) is just an ensemble of random samples without any guarantee for the representative of the input space.

The Gaussian process emulator used in this work is not suitable for models with hundreds of dimensions. A large number of training samples would make it difficult when computing the inverse correlation matrix. In those cases, the implementation of Neural networks [142] can be considered. Neural networks can be an alternative approach for interpolating an expensive simulator with hundreds of input dimensions to reduce the

running cost. In addition, Principal component analysis [143] can convert the high dimensional inputs into a smaller set of components remaining the largest impact on the model output, so that the dimensionality can be reduced.

Considering that an exhaustive exploration for continuous input space is not possible, SSHM cannot ensure that all fitting regions are found. This can be checked by increasing the size of the samples. If some new non-implausible regions appear, more samples should be used until enough regions are found. Information from experts can be considered that they may provide an approximate volume for the fitting domain.

To avoid excluding possible non-implausible domain, the implausibility threshold considers various sources of uncertainty and allows 3 times standard deviation from the given observation. This could be related to trying to avoid the type I error in statistics, which is the rejection of a true hypothesis. The resulting non-implausible samples can be verified by the simulator to detect whether they can provide the desired match between the model output and the observation. This could be related to trying to avoid the type II error in statistics, which is failing to reject a false hypothesis. In statistical hypothesis test theory, increasing the sample size can also reduce both types of error [144].

In the uncertainty quantification for an expert estimated correlation matrix, expert estimates are assumed close to the real correlations, but if they are not, the proposed algorithms would lead to a valid correlation matrix, faraway from the expert-estimated matrix.

7.3 Future work

There are several future research directions. First, this dissertation only studied independent outputs. Considering applications of HM calibrating multiple dependent physical quantities ([33, 145, 146]), SSHM could adopt multivariate emulation in the future. Second, disconnected non-implausible domain may have different behaviors of the model output, hence different emulators should be trained in disconnected regions, with different priors and smoothing parameters. Third, for a very high dimensional input space,

it is difficult to fully describe all uncertainties and make robust priors. In this case, a full Bayes approach is not feasible, instead that a Bayes linear approach [147, 148] could be considered. It only makes inference about the expectation and variance of the model output without a full probabilistic distribution, using the training data's expectation, variance and covariance. Bayes linear analysis has proven to be sufficient for emulating in many applications [45, 149, 150]. Fourth, for quantifying the uncertainty of expert-estimated correlations, the fact that empirical estimates are also only approximate was ignored. The difference between the actual correlation and its empirical estimate could be determined via the sample size, in other words the standard deviation of the approximate estimate. We hope that this work could spark future improvements of model calibration methods and relevant research field.

List of Publications

Journal article:

- Z. T. Gong, F. A. DiazDelaO, and M. Beer. Bayesian model calibration using Subset Simulation. *Reliability Engineering and System Safety.* (Submitted)

Conference Papers:

- Z. T. Gong, F. A. DiazDelaO, and M. Beer. Bayesian model calibration using Subset Simulation. *In Risk, Reliability and Safety: Innovating Theory and Practice - Proceedings of the 26th European Safety and Reliability Conference, ESREL,* Glasgow, UK, September 25 - September 29, 2016.
- Z. T. Gong, F. A. DiazDelaO, and M. Beer. Sampling Schemes for History Matching Using Subset Simulation. *Proceedings of the 2nd International Conference on Uncertainty Quantification in Computational Sciences and Engineering, UNCECOMP,* Rhodes Island, Greece, June 15 - June 17, 2017.
- M. Beer, Z. Gong, F. A. DiazDelaO, and V. Kreinovich. How Accurate Are Expert Estimations of Correlation? *Proceedings of the 2017 IEEE Symposium on Computational Intelligence for Engineering Solutions, CIES,* Honolulu, Hawaii, November 27 - December 1, 2017.
- M. Beer, Z. Gong, I. Neumann, S. Sriboonchitta, and V. Kreinovich. What if we do not know correlations? *Studies in Computational Intelligence - Proceedings of Econometrics for Financial Applications, ECONVN,* Ho Chi Minh City, Vietnam, January 15 - January 16, 2018.

Bibliography

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 5, 4.6.1, 4.13

[2] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, (4):409–423, 1989. 1.1, 1.1, 2.5.1

[3] S. R. Nassif, A. J. Strojwas, and S. W. Director. FABRICS II: A statistically based IC fabrication process simulator. *IEE Trans. Computer-Aided Design CAD-3*, pages 40–46, 1984. 1.1

[4] R. J. Kee, J. F. Grcar, M. D. Smooke, and J. A. Miller. A fortran program for modelling steady laminar one-dimensional premixed flames. *Sandia Report SAND85-8240, National Technical Information Service, Springfield, Va. 22161*, 1985. 1.1

[5] H. Wang, X. You, A. V. Joshi, S. G. Davis, A. Laskin, F. Egolfopoulos, and C. K. Law. High-temperature combustion reaction model of H2/CO/C1-C4 compounds. *USC Mech Version II*, 2007. 1.1

[6] T. Bjerga, T. Aven, and E. Zio. An illustration of the use of an approach for treating model uncertainties in risk assessment. *Reliability Engineering and System Safety*, 125:46–53, 2014. 1.1

[7] P. S. Laplace. *Théorie analytique des probabilités*. Mme Ve Courcier, 1820. 1.1

[8] M. B. Beck. Water quality modelling: a review of the analysis of uncertainty. *Water Resources Res*, pages 1393–1442, 1987. 1.1

[9] D. P. Gallegos and E. J. Bonano. Consideration of uncertainty in the preformance assessment of radioactive waste disposal from an international regulatory perspective. *Reliab. Engng System Safety*, pages 111–123, 1993. 1.1

[10] E. Zio and G. E. Apostolakis. Two methods for the structured assessment of model uncertainty by experts in performance assessments of radioactive waste repositories. *Reliab Eng Syst Saf*, pages 225–241, 1996. 1.1

[11] T. Nilsen and T. Aven. An exploration of alternative approaches to the representation of uncertainty in model predictions. *Reliab Eng Syst Saf*, pages 309–317, 2003. 1.1

[12] J. C. Helton, H. D. Johnson, and W. L. Oberkampf. Models and model uncertainty in the context of risk analysis. *Reliab Eng Syst Saf*, pages 171–185, 2004. 1.1

[13] J. McFarland and S. Mahadevan. Calibration and uncertainty analysis for computer simulations with multivariate output. *AIAA Journal*, pages 1253–1265, 2008. 1.1

[14] ASME. Guide for verification and validation in computational solid mechanics. pages 1–15, 2007. 1.1

[15] T. H. Naylor and J. M. Finger. Verification of computer simulation models. *Management Science*, 14:B–92–B–101, 1967. 1.1

[16] J. C. Refsgaard. Parameterisation, calibration and validation of distributed hydrological models. *Journal of Hydrology*, 198:69–97, 1997. 1.1

[17] J. C. Refsgaard. Automatic calibration of conceptual rainfall-runoff models; sensitivity to calibration data. *Journal of Hydrology*, 181:23–48, 1996. 1.1

[18] S. Sankararaman and S. Mahadevan. Integration of model verification, validation, and calibration for uncertainty quantification in engineering systems. *Reliability Engineering & System Safety*, 138:194–209, 2015. 1.1

[19] D. Taylor, A. Pandya, and D. Thompson. Fundamentals of model calibration: theory & practice. *ISPOR 17th Annual International Meeting, Washington, DC, USA*, 2012. 1.1

[20] S. B. Vardeman. What about the other intervals? *The American Statistician*, 46: 193–197, 1992. 1.1

[21] L. Wang, M. D. Gordon, and J. Zhu. Regularized least absolute deviations regression and an efficient algorithm for parameter tuning. *Proceedings of the Sixth International Conference on Data Mining*, 57:690–700, 2006. 1.1

[22] A. B. owen. *Empirical likelihood*. London: Chapman & Hall/Boca Raton, FL: CRC Press, 2001. 1.1

[23] Z. Ullah. Parallel computations in nonlinear solid mechanics using adaptive finite element and meshless methods. *Engineering Computations: International Journal for ComputerAided Engineering and Software*, 33:1161–1191, 2015. 1.1

[24] M. N. Akhtar, M. H. Durad, A. Usman, and Aman ur Rehman. Steady state heat transfer using galerkin finite element method. *2015 Fourth International Conference on Aerospace Science and Engineering, Islamabad*, pages 1–6, 2015.

[25] P. M. Roehm. Minimizing run time of finite element analyses: applications in conformable CNG tank modeling., 2017. 1.1, 2.1

[26] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker. Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, (416):953–965, 1991. 1.1, 2.5.1

[27] M. D. Morris, T. J. Mitchell, and D. Ylvisaker. Bayesian design and analysis of computer experiments: use ofderivatives in surface prediction. *Technometrics*, 35: 243–255, 1993.

[28] R. A. Bates, R. J. Buck, E. Riccomagno, and H. P. Wynn. Experimental design and observation for largesystems. *J. R. Statist. Soc. B*, 58:77–94, 1996. 1.1

[29] M. C. Kennedy and A. O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87:1–13, 2000. 1.1

[30] J. Oakley. Bayesian uncertainty analysis for complex computer codes., 1999. 1.1, 2.1, 2.2.2, 2.5.2, 2

[31] J. Aitchison and I. R. Dunsmore. *Statistical Prediction Analysis*. Cambridge University Press, England, 1975. 1.1

[32] M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, (3):425–464, 2001. 1.1, 2.2.3

[33] D. Higdon, J. Gattiker, B. Williams, and M. Rightley. Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, (482):570–583, 2008. 1.1, 7.3

[34] J. McFarland and S. Mahadevan. Multivariate significance testing and model calibration under uncertainty. *Comput. Methods Appl. Mech. Engrg.*, 197:29–32, 2008. 1.1

[35] M. Ramin and G. B. Arhonditsis. Bayesian calibration of mathematical models: Optimization of model structure and examination of the role of process error covariance. *Ecological Informatics*, 18:107–116, 2013. 1.1

[36] G. B. Arhonditsis, S. S. Qian, C. A. Stow, E. C. Lamon, and K. H. Reckhow. Eutrophication risk assessment using bayesian calibration of process-based models: application to a mesotrophic lake. *Ecological Modelling*, 208:215–229, 2007. 1.1

[37] J. Ray, S. Lefantzi, S. Arunajatesan, and L. Dechant. Bayesian parameter estimation of a k-ε model for accurate jet-in-crossflow simulations. *AIAA Journal*, 54: 2432–2448, 2016. 1.1

[38] I. Andrianakis and P. Challenor. The effect of the nugget on gaussian process emulators of computer models. *Computational Statistics and Data Analysis*, 40: 4215–4228, 2012. 1.1, 1.1, 1.2, 2.1, 2.2

[39] F. Anterion. History matching: a one day long competition: classical approaches versus gradient method. *First International Forum On Reservoir Simulation*, 1998. 1.1

[40] P. Craig, M. Goldstein, A. Seheult, and J. Smith. Bayes linear strategies for matching hydrocarbon reservoir history. *Bayesian Statistics*, pages 69–95, 1996. 1.1, 1.2

[41] R. Baker. Streamline technology, reservoir history matching and forecasting; its success, limitations, and future. *Journal of Canadian Petroleum Technology*, 40: 23–27, 2001.

[42] M. Alfi and S. A. Hosseini. Integration of reservoir simulation, history matching, and 4d seismic for CO2-EOR and storage at Cranfield, Mississippi, Usa. *Fuel*, 175: 116–128, 2016. 1.1

[43] R. Tokmakian, P. Challenor, and Andrianakis Y. On the use of emulators with extreme and highly nonlinear geophysical simulators. *Journal of Atmospheric And Oceanic Technology*, 29:1704–1715, 2012. 1.1

[44] I. Vernon, M. Goldstein, and R. Bower. Galaxy formation: Bayesian history matching for the observable universe. *Statistical Science*, 29(1):81–90, 2014. 1.1, 2.3, 2.6, 3

[45] I. Vernon, M. Goldstein, and R. G. Bower. Galaxy formation: a bayesian uncertainty analysis. *Bayesian Analysis*, 5(4):619–669, 2010. 1.1, 1.1, 2.1, 4.1, 4.5.2, 7.3

[46] D. Williamson and A. T. Blaker. Evolving Bayesian emulators for structured chaotic time series, with application to large climate models. *Society for Industrial and Applied Mathematics*, pages 1–28, 2014. 1.1, 1.2

[47] D. Williamson and I. Vernon. Efficient uniform designs for multi-wave computer experiments. arxiv:1309.3520 [stat.me]. 2013. 1.1, 2.1, 2.7.2, 4.4, 4.4.2, 4.4.2

[48] I. Andrianakis, I. R. Vernon, N. McCreesh, T. J. McKinley, J. E. Oakley, R. N. Nsubuga, M. Goldstein, and R. G. White. Bayesian history matching of complex

infectious disease models using emulation: a tutorial and a case study on HIV in Uganda. *Plos Computational Biology*, 11(1), 2015. 1.1, 2.2.3, 2.6, 2.7.2, 2

[49] G. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic.* Prentice Hall, Upper Saddle River, New Jersey, 1995. 1.1, 6.2

[50] H. T. Nguyen and E. A. Walker. *A First Course in Fuzzy Logic.* Chapman and Hall/CRC, Boca Raton, Florida, 2006.

[51] L. A. Zadeh. Fuzzy set. *Information and Control*, pages 338–353, 1965. 1.1, 6.1, 6.2

[52] S. K. Au and J.L. Beck. Estimation of small failure probabilities in high dimensions by Subset Simulation. *Probabilistic Engineering Mechanics*, 16(4):263–277, 2001. 1.2, 2.7.2, 3.1, 3.2, 3.3.1, 5.3

[53] S. J. M. Ester, H. P. Kriegel, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proc. of 2nd International Conference on Knowledge Discovery*, pages 226–231, 1996. 1.2, 2.7.1, 4.6.1, 4.6.1

[54] P.S. Craig, M. Goldstein, A.H. Seheult, and J.A. Smith. Pressure matching for hydrocarbon reservoirs: a case study in the use of Bayes linear strategies for large computer experiments (with discussion), in gatsonis et al. (eds). *Case Studies in Bayesian Statistics*, III:37–93, 1997. 2.1

[55] E. P. George. Science and statistics. *Journal of the American Statistical Association*, 71:791–799, 1976. 2.2.3

[56] J. Brynjarsdóttior and A. O'Hagan. Learning about physical parameters: the importance of model discrepancy. *Inverse Problems*, 2014. 2.2.3

[57] D. Higdon, M. C. Kennedy, J. Cavendish, and R. Ryne. Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing*, 26(2):448–466, 2004. 2.2.3

[58] T. S. Jr. and K. Burrage. Computational and structural biotechnology journal. *Biometrika*, 12:14–25, 2014. 2.2.4

[59] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, (2):239–245, 1979. 2.4.1

[60] M. D. Morris and T. J. Mitchell. Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, 43(3):381–402, 1995. 2.4.1, 2.4.1, 2.4.1

[61] A. I. J. Forrester, A. Sobester, and A. J. Keane. *Engineering design via surrogate modelling: a practical guide.* Chichester: J. Wiley, 2008. 2.4.1, 2.7.1, 4.5.2

[62] G. E. P. Box. Evolutionary operation: a method for increasing industrial productivity. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, (2): 81–101, 1957. 2.4.1, 2.5.2

[63] M.E. Johnson, L.M. Moore, and D. Ylvisaker. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26(2):131–148, 1990. 2.4.1

[64] J. L. Loeppky, J. Sacks, and W. J. Welch. Choosing the sample size of a computer experiment: a practical guide. *Technometrics*, (4):366–378, 2009. 2.4.2

[65] G. Wahba. *Spline models for observational data.*, volume 59 of *CBMS-NSF Regional Conference series in applied mathematics*. Philadelphia: Society for Industrial & Applied Maths, 1990. 2.5.1

[66] J. E. Oakley and A. O'Hagan. Probabilistic sensitivity analysis of complex models: a bayesian approach. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, (3):751–770, 2004.

[67] A. O'Hagan. Bayesian analysis of computer code outputs: a tutorial. *Reliability Engineering and System Safety*, 91:1290–1300, 2006. 2.5.1

[68] F. A. DiazDelaO, S. Adhikari, E.I. Saavedra Flores, and M.I. Friswell. Stochastic structural dynamic analysis using bayesian emulators. *Computers and Structures*, 120:24–32, 2013. 2.5.1

[69] J. Ghosh, J. E. Padgett, and L. Dueñas-Osorio. Surrogate modeling and failure surface visualization for efficient seismic vulnerability assessment of highway bridges. *Probabilistic Engineering Mechanics*, 34:189–199, 2013. 2.5.1

[70] E.I. Saavedra Flores, F. A. DiazDelaO, M. I. Friswell, and J. Sienz. A computational multi-scale approach for the stochastic mechanical response of foam-filled honeycomb cores. *Composite Structures*, 94:1861–1870, 2012. 2.5.1

[71] L. Pichler, H.J. Pradlwarter, and G.I. Schuëller. A mode-based meta-model for the frequency response functions of uncertain structural systems. *Computers and Structures*, 87:332–341, 2009. 2.5.1

[72] B. Minasny and A. B. McBratney. The Matern function as a general model for soil variograms. *Geoderma*, 128(3-4):192–207, 2005. 2.5.1, 4.5.1

[73] Antony Overstall and David Woods. Multivariate emulation of computer simulators: model selection and diagnostics with application to a humanitarian relief model. *arXiv:1506.04489v2 [stat.ME]*, 2015. 2.5.2

[74] H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, pages 453–461, 1946. 2.5.2

[75] I. J. Myung and D. J. Navarro. *Information matrix.* Encyclopedia of Statistics in Behavioral Science, 2005. 2.5.2

[76] O. Roustant, D. Ginsbourger, and Y. Deville. DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51(1):1–55, 2012. 2.5.4

[77] L. S. Bastos and A. O'Hagan. Diagnostics for gaussian process emulators. *Technometrics*, (4):425–438, 2009. 2.5.4, 2.5.4

[78] P. C. Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):49–55, 1936. 2.5.4

[79] A. Pepelyshev and J. Oakley. On the choice of correlation function and cross-validation for gaussian processes. *MUCM technical reprot*, 2009. 2

[80] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning.* Adaptive computation and machine learning. Cambridge, Mass.: MIT Press, 2006. 2

[81] F. Pukelsheim. The three sigma rule. *The American Statistician*, (2):68–95, 1994. 2.6, 4.5.1

[82] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, pages 345–383, 2001. 2.7.1

[83] C. J. Geyer. Markov chain monte carlo maximum likelihood. *Computing Science and Statistics*, pages 156–163, 1991. 2.7.2, 3.2

[84] F Liang and W. H. Wong. Real-parameter evolutionary monte carlo with applications to bayesian mixture models. *J. Am. Stat. Assoc*, pages 653–666, 2001. 2.7.2

[85] J. A. Cumming and M. Goldstein. Bayes linear uncertainty analysis for oil reservoirs based on multiscale computer experiments. *The Oxford Handbook of Applied Bayesian Analysis*, pages 241–270, 2010. 1

[86] E. Zio. Reliability engineering : old problems and new challenges. *Relia Eng Syst Saf*, 94:125–141, 2009. 3.1

[87] S. K. Au and Y. Wang. *Engineering Risk Assessment with Subset Simulation.* Wiley, 2014. 3.1, 3.1, 3.1, 3.1, 3.2, 3, 3.3.1, 4

[88] D. L. Logan. *A first course in the finite element method.* Cengage Learning, USA, 2012. 3.1

[89] I. Doltsinis. Perturbation-based stochastic FE analysis and robust design of inelastic deformation processes. *Computer Methods in Applied Mechanics and Engineering*, 195:2231–2251, 2006. 3.1

[90] T. P. Chang. Stochastic FEM on nonlinear vibration of fluid-loaded double-walled carbon nanotubes subjected to a moving load based on nonlocal elasticity theory. *Composites Part B: Engineering*, 54:391–399, 2006.

[91] C. Lang, A. Doostan, and K. Maute. Extended stochastic FEM for diffusion problems with uncertain material interfaces. *Computational Mechanics*, 51:1031–1049, 2013. 3.1

[92] N. Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, (247):335–341, 1949. 3.1

[93] N. Metropolis. The beginning of the monte carlo method. *Los Alamos Science*, pages 125–130, 1987. 3.1

[94] E. Zio, P. Baraldi, and E. patelli. Assessing the availability of an offshore installation by monte carlo simulation. *Int J Press Vessel Piping*, 83:312–320, 2006. 3.1

[95] O. Ditlevsen and H. O. Madsen. *Structural Reliability Methods.* Chichester: John Wiley & Sons, 1996. 3.1

[96] H. O. Madsen, S. Krenk, and N. C. Lind. *Methods of structural safety.* Dover Publications, Inc., Mineola, N. Y., 2006.

[97] R. Melchers. *Structural reliability analysis and prediction.* Chichester: John Wiley & Sons, 1999. 3.1

[98] S. Engelund and R. Rackwitz. A benchmark study on importance sampling techniques in structural reliability. *Structural Safety*, pages 255–276, 1993. 3.1

[99] G. I. Schuëller and R. Stix. A critical appraisal of methods to determin failure probabilites. *Struct Safety*, 4:293–309, 1987. 3.1

[100] R. E. Melchers. Importance sampling in structural systems. *Struct Safety*, 6:3–10, 1989.

[101] M. Hohenbichler and R. Rackwitz. Improvement of second-order reliability estimates by importance sampling. *J Engng mech, ASCE*, 114(12):2195–2198, 1988.

[102] A. Der-Kiureghian and T. Dakessian. Multiple design points in first and second-order reliability. *J Engng mech, ASCE*, 20:37–49, 1998. 3.1

[103] G. I. Schuëller, H. J. Pradlwarter, and P. Koutsourelakis. A critical appraisal of reliability estimation procedures for high dimensions. *Probabilistic Engineering mechanics*, pages 463–474, 2004. 3.1

[104] S. K. Au and E. Patelli. Rare event simulation in finite-infinite dimensional space. *Reliability Engineering and System Safety*, 148:67–77, 2016. 3.2, 3.3.5, 3.4, 3.4

[105] D. A. Schum. *The evidential foundations of probabilistic reasoning.* Northwestern University Press, 1994. 3.2

[106] H. E. Klugh. *Statistics: the essentials for research.* Psychology Press, 2013. 3.2

[107] J. Ching, S. K. Au, and J. L. Beck. Reliability estimation for dynamical systems subject to stochastic excitation using subset simulation with splitting. *Computer Methods in Applied Mechanics and Engineering*, 194(Special Issue on Computational Methods in Stochastic Mechanics and Reliability Analysis):1557–1579, 2005. 3.3

[108] E. Zio and N. Pedroni. How to effectively compute the reliability of a thermal—hydraulic nuclear passive system. *Nuclear Engineering and Design*, 241:310–327, 2011. 3.3

[109] W. K. Hasting. Monte Carlo sampling methods using Markov Chains and their applications. *Biometrica*, 57:97–109, 1970. 3.3.1

[110] N. Metropolis, A. W. Rosenbluth, M. N. Rosenblut, A. H. Teller, and E. Teller. Tquation of state calculations by fast computing machines. *J. Chem. Phys.*, 21: 1087–1092, 1953. 3.3.1

[111] K. M. Zuev. Understanding the subset simulation method for rare event estimation. *Technical note*, 2015. 3.3.1

[112] S. S. Chen and R. A. Gopinath. Gaussianization. *Technical Report, IBM T. J. Watson Research Center*, 2000. 3.3.1

[113] I. Papaioannou, W. Betz, K. Zwirglmaier, and D. Straub. MCMC algorithms for subset simulation. *Probabilistic Engineering Mechanics*, 41:89–103, 2015. 3.3.1, 3.3.4, 2, 3.3.5

[114] A. M. Santoso, K. K. Phoon, and S. T. Quek. Modified Metropolis–Hastings algorithm with reduced chain correlation for efficient subset simulation. *Probabilistic Engineering Mechanics*, 26:331–341, 2011. 3.3.2, 3.3.2, 4

[115] F. Miao and M. Ghosn. Modified subset simulation method for reliability analysis of structural systems. *Struct. Saf.*, 33:233–241, 2011. 3.3.3

[116] G. O. Roberts, A. Gelman, and W.R. Gilks. Weak convergence and optimal scaling of random walk metropolis algorithms. *Ann. Appl. Probab.*, 7:110–120, 1997. 3.3.4

[117] G. O. Roberts and J. S. Rosenthal. Optimal scaling for various metropolis–hastings algorithms. *Stat. Sci.*, 16:351–367, 2001. 3.3.4

[118] E. Patelli and S. K. Au. Efficient monte carlo algorithm for rare failure event simulation. *The 12th International Conference on Applications of Statistics and Probability in Civil Engineering held in Vancouver, Canada on July 12-15*, 2015. 1

[119] T. Williams and C. Kelley. Gnuplot demo script: contours.dem. `http://gnuplot.sourceforge.net/demo/contours.html`, 2015. 4.4.1

[120] S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. 2016. 4.5.2

[121] MathWorks. Graphics. `http://uk.mathworks.com/help/matlab/ref/peaks.html`, 2016. 4.6.2

[122] M. Darecki, C. Edelstenne, T. Enders, E. Fernandes, P. Hartman, J. Herteman, M. Kerkloh, I. King, P. Ky, M. Mathieu, G. Orsi, G. Schotman, C. Smith, and J. Wörner. Flightpath 2050, europeś vision for aviation, maintaining global leadership & serving society's needs. *Report of the High Level Group on Aviation Research*, pages 1–28, 2011. 5.1

[123] M. D. Guenov, M. Nunez, A. Molina-Cristóbal, V. Sripawadkul, V. Datta, and A. Riaz. Composition, management, and exploration of computational studies at early design stage. *Computational Intelligence in Aerospace Sciences*, pages 415–460, 2014. 5.1

[124] M. D. Guenov, A. Nunez, M.and Molina-Cristóbal, V. Datta, and A. Riaz. AirCADia-An interactive tool for composition and exploration of aircraft computational studies at early design stage. *Proceedings of the 29th Congress of the International Council of the Aeronautical Sciences, St. Petersburg, Russia, September 7th–12th*, 2014. 5.1

[125] M. Rudemo. Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, pages 65–78, 1982. 5.3

[126] C. R. Johnson. Positive definite matrices. *Amer. Math. Monthly*, 77:259–264, 1970. 6.1

[127] M. Marcus and H. Minc. *Introduction to linear algebra.* New York: Dover, 1988.

[128] I. V. Lindell. *Methods for electromagnetic field analysis.* New York: Clarendon press, 1992. 6.1

[129] D. Klaua. An early approach toward graded identity and graded membership in set theory. *Fuzzy Sts and Systems*, 161(18):2369–2379, 1965. 6.1

[130] A. N. Blaira, B. M. Ayyub, and W. J. Bender. Fuzzy stochastic risk-based decision analysis with the mobile offshore base as a case study. *Marine Strucures*, 14(1-2): 69–88, 2001. 6.1

[131] N. J. Higham. Computing the nearest correlation matrix—a problem from finance. *IMA Journal of Numerical Analysis*, 22(3):329–343, 2002. 6.2, 6.3

[132] M. Stein, M. Beer, and V. Kreinovich. Bayesian approach for inconsistent information. *Information Sciences*, pages 96–111, 2013. 6.2

[133] R. Borsdorf. A newton algorithm for the nearest correlation matrix. *M.Sc. Thesis, University of Manchester*, 2007. 6.3

[134] R. Borsdorf. Structured matrix nearness problems: Theory and applications. *PhD Dissertation, University of Manchester*, 2012.

[135] R. Borsdorf and N. J. Higham. A preconditioned newton algorithm for the nearest correlation matrix. *IMA Journal of Numerical Analysis*, pages 94–107, 2010.

[136] N. J. Higham R. Borsdorf and M. Raydan. Computing a nearest correlation matrix with factor structure. *SIAM Journal on Matrix Analysis and Applications*, pages 2603–2622, 2010.

[137] C. Lucas. Computing nearest covariance and correlation matrices. *M.Sc. Thesis, University of Manchester*, 2001.

[138] H. Qi and D. Sun. A quadratically confergent newton method for computing the nearest correlation matrix. *SIAM Journal on Matrix Analysis and Applications*, pages 360–385, 2006. 6.3

[139] W. J. Morokoff and R. E. Caflisch. Quasi-monte carlo integration. *J. Comput. phys.*, 2:218–230, 1995. 7.2

[140] I. M. Sobol. Distribution of points in a cube and approximate evaluation of integrals. *Comput. Maths. Maths. Phys.*, 7:86–112, 1967. 7.2

[141] L. Kocis and W. J. Whiten. Computational investigations of low-discrepancy sequences. *Transactions on Mathematical Software*, 23:266–294, 1967. 7.2

[142] S. Haykin. *Neural networks: a comprehensice foundation.* ACM, 1998. 7.2

[143] H. Abdi and L. J. Williams. Principal component analysis. *Computational Statistics*, 2:433–459, 2010. 7.2

[144] W. H. Ang, A. H.and Tang. *Probability concepts in engineering: emphasis on applications in civil and environmental engineering.* Hoboken, N.J.: John Wiley, Sons, 2007. 7.2

[145] S. Conti and A. O'Hagan. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140:640–651, 2010. 7.3

[146] J. Rougier. Efficient emulators for multivariate deterministic functions. *Journal of Computational and Graphical Statistics*, (4):827, 2008. 7.3

[147] M Goldstein. *Bayes linear analysis.* Encyclopaedia of Statistical Sciences, Chichester, Wiley, 1999. 7.3

[148] M. Goldstein and D. A. Wooff. *Bayes linear statistics: Theory and Methods.* Bayes Linear Statistics, Chichester, Wiley, 2007. 7.3

[149] L. F. S. Rodrigues, I. Vernon, and R. G. Bower. Constraints on galaxy formation models from the galaxy stellar mass function and its evolution. arxiv:1609.06922v3 [astro-ph.ga]. pages 1–19, 2016. 7.3

[150] M. J. Bayarri, R. Paulo J. O. Berger, J. Sacks, J. A. Cafeo, J. Cavendish, C. H. Lin, and J. Tu. A framework for the validation of computer models. *Technometrics*, pages 138–154, 2007. 7.3